Technische Universität München
Fakultät für Informatik

Lehrstuhl für Effiziente Algorithmen

Ranking and Ordering Problems of Spanning Trees

Matthias Baumgart

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. H. Seidl
Prüfer der Dissertation:		
	1.	Univ.-Prof. Dr. E. W. Mayr
	2.	Univ.-Prof. Dr. F. J. Esparza Estaun

Die Dissertation wurde am 23.04.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 06.11.2009 angenommen.

Document Classification according to ACM CCS (1998)

Categories and subject descriptors:

`F.2.2` [Analysis of Algorithms and Problem Complexity]:
- $\triangleright$ Nonnumerical Algorithms and Problems
  - $\triangleright$ Computations on Discrete Structures

`G.2.1` [Discrete Mathematics]:
- $\triangleright$ Combinatorics
  - $\triangleright$ Combinatorial Algorithms, Counting Problems, Permutations and Combinations

`G.2.2` [Discrete Mathematics]:
- $\triangleright$ Graph Theory
  - $\triangleright$ Graph Algorithms, Graph Labeling, Trees

Abstract

Each spanning tree T of an undirected graph $G = (V, E)$ is represented by a vertex in the tree graph of G. Two of these 'spanning tree' vertices are connected by an edge if and only if the corresponding spanning trees are related by an edge swap. This definition can also be extended to undirected weighted graphs $G = (V, E, w)$ and weighted spanning trees. A couple of questions have arisen regarding these tree graphs. Some conjectures regarding different weighted spanning trees were proposed by Kano. Mayr and Plaxton who proved one of Kano's conjectures, formulated a unifying conjecture concerning bispanning graphs. The edge set of these graphs consists of two edge-disjoint spanning trees which we label with P and Q. More precisely, we consider weighted bispanning graphs $B = (V, P, Q, w)$ restricted to weight functions w such that $w(P) < w(Q)$ and Q is the only spanning tree with weight $w(Q)$. Then, it is conjectured that there are $|V|-1$ spanning trees with pairwise different weights where each of them is smaller than $w(Q)$. We are able to prove this claim if we restrict ourselves to specially weighted or specially structured bispanning graphs. We consider weighted bispanning graphs $B = (V, P, Q)$ such that both spanning trees, P and Q, have unique weights. Furthermore, we analyze the structure of a bispanning graph using some ideas from matroid theory. Based on these findings, we formulate a refinement of Mayr and Plaxton's conjecture. We show that it might be possible to count only spanning trees which define a new partition of a bispanning graph.

Strongly related to paths in tree graphs are so-called base orderings, which are also defined in the context of matroids. It is known that for each bispanning graph $B = (V, P, Q)$, there exists a cyclic base ordering of P and Q. A cyclic base ordering is an ordering of $P = \{p_1, \ldots, p_m\}$ and $Q = \{q_1, \ldots, q_m\}$ such that any m cyclically consecutive edges in the sequence $q_1 \ldots q_m\, p_1 \ldots p_m$ form a spanning tree. In the context of spanning trees of a graph, i.e., for graphic matroids, we propose a stronger property of such orderings which we will call subsequence-interchangeable base orderings. We claim that each bispanning graph has an ordering that achieves this stronger property. Although we cannot prove this in general, we present a variety of bispanning graphs that have such an ordering. In particular, we present an algorithm to construct a subsequence-interchangeable base ordering for each partition of the wheel graph W_n into two spanning trees. Analogous to the problem of counting weighted spanning trees, we identify a subclass of bispanning graphs to which the problem can be reduced. Subsequently, we describe an approach to using subsequence-interchangeable base orderings to solve the above counting problem.

Finally, we consider a network sparsification problem called GAST: Compute a spanning tree of a connected undirected graph $G = (V, E)$ that minimizes the sum of distance differences of all vertex pairs $u, v \in V$ which are connected by an edge $\{u, v\} \in E$. We show that the decision variant of this problem is $\mathcal{NP}$-complete with respect to the L_p norm for arbitrary $p \in \mathbb{N}$. For the reduction, we use the $\mathcal{NP}$-complete problem 2-Hitting Set, which is more commonly known as Vertex Cover. For the L_1 norm, we give a reduction to the Minimum Fundamental Cycle Basis Problem (Min-FCB). If Min-FCB is approximable within $\rho > 1$ then GAST with respect to L_1 is approximable within 3ρ. For arbitrary graphs, the approximation ratio is $\mathcal{O}(\log^2 n \log\log n)$.

Acknowledgments

First and foremost, I am deeply grateful to my supervisor Ernst W. Mayr for his support throughout the time of research and writing this thesis. Many thanks go to all present and former colleagues at the *Chair for Efficient Algorithms* for providing a very pleasant working environment. In particular, I thank Stefan Eckhardt for his generous help on many occasions. Special thanks go to Hanjo Täubig and Michael Schnupp for numberless discussions about spanning trees and bispanning graphs. Last but not least, I thank my parents for all their support during the time of writing this thesis.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Outline 5

2 Preliminaries **7**
2.1 Graphs, Trees, and Tree Graphs 7
2.2 Bispannable and Bispanning Graphs 11
2.2.1 Construction of Bispanning Graphs 14
2.2.2 Computing a Partition into two Spanning Trees 20

3 Ranking of Weighted Spanning Trees **25**
3.1 Kano's Conjectures 25
3.2 Assuming Singularity of P 28
3.3 An Analysis using Matroid Theory 32
3.3.1 Preliminaries 32
3.3.2 Transversal and Strongly Base Orderable Matroids 35

4 Partitioning Bispanning Graphs **41**
4.1 Introduction 41
4.2 Strictly 2-Edge-Connected Bispanning Graphs 43
4.3 Decomposition of Bispanning Graphs with Cut Vertices 44
4.4 General Decomposition of Bispanning Graphs 52
4.5 Partitioning the K_4 54

5 Subsequence-Interchangeable Base Orderings **59**
5.1 Introduction 59
5.2 Cyclic Base Orderings 60
5.3 Subsequence-Interchangeable Base Orderings 64
5.3.1 Bottom-Up Approach 68
5.3.2 Top-Down Approach 79
5.3.3 The Class of Wheel Graphs 80
5.4 Concluding Remarks 84

6 Graph-Approximating Spanning Trees **87**
6.1 Introduction 87
6.2 The 2-Hitting-Set Gadget 88
6.3 Graph-Approximating Spanning Trees 91
6.4 Approximating GAST with respect to L_1 96

7 Conclusion **99**

Bibliography **101**

Index **111**

List of Figures

2.1 Contracting an edge. 8
2.2 A minor of a graph. 8
2.3 Two spanning trees related by an edge swap. 9
2.4 A graph and its tree graph 10
2.5 A bispannable graph and two different partitions. 11
2.6 Connectivity of bispanning graphs. 12
2.7 Operation double-leaf attachment. 15
2.8 Operation edge-split. 15
2.9 Reversing an edge-split. 17
2.10 Different construction sequences for the same graph. 18
2.11 Using edge-splits to make a composite bispannable graph atomic. . . . 19
2.12 Different graphs for the Shannon switching game. 23

3.1 Constructing new classes of spanning trees (a). 30
3.2 Constructing new classes of spanning trees (b). 31
3.3 Bipartite graph representation of a transversal matroid. 35
3.4 K_4 and C_5^2. 38
3.5 The cycle matroid is strongly base orderable but not transversal. 39

4.1 A weighted bispanning graph with $1 = \sigma(B, Q) < \sigma(B, P)$. 41
4.2 Transforming strictly 2-edge-connected bispanning graphs. 43
4.3 Constructing new classes of partition spanning trees (a). 45
4.4 Constructing new classes of partition spanning trees (b). 46
4.5 Two bispanning graphs joined by a cut vertex. 48
4.6 Combinations of partition spanning trees if $w(P_1) < w(Q_1)$. 50
4.7 Combinations of partition spanning trees if $w(P_1) > w(Q_1)$. 51
4.8 Constructing a minor isomorphic to K_4. 54
(a) There are two chords with a common vertex. 54
(b) No two chords have a common vertex. 54
4.9 A decomposition of K_4 into two disjoint spanning trees. 55
4.10 All partition spanning trees of the complete graph K_4. 56

5.1 Reversing an edge-split of P. 62
5.2 All paths of the K_4's tree graph corresponding to SIBOs. 66

5.3 The K_4's tree graph restricted to partition spanning trees. 67
5.4 Splitting an edge of P. 69
5.5 Splitting an edge of Q. 70
5.6 Splitting a 'middle' edge (a). 74
5.7 Splitting a 'middle' edge (b). 74
5.8 Splitting a 'middle' edge (c). 76
5.9 Splitting a 'middle' edge (d). 76
5.10 Restricted edge-split operation. 77
5.11 Using only restricted edge-splits and double-leaf attachments 78
5.12 The wheel W_n. 80
5.13 Processing a partition of the Wheel W_7. 82

6.1 Graph representation $G(\mathcal{C}, \mathcal{S})$ of a 2HS instance. 89
6.2 Extended graph representation of a 2HS instance. 93
6.3 Extension paths of a clause path edge $\{u, v\}$. 94

Chapter 1

Introduction

1.1 Motivation

Minimum Spanning Trees

A variety of real-world problems can be modeled as network or graph problems. In this context, the importance and impact of algorithmic graph theory and discrete mathematics can never be overrated. For example, the road network can be mapped to a directed or undirected graph in a natural way. Another example is the problem of assigning a number of operators with different technical knowledge to a set of different machines. This can be modeled as a bipartite graph. Then, the task is to develop efficient graph algorithms to solve various objectives, e.g., we are looking for efficient algorithms to compute a shortest path, a maximum flow, or a maximum matching.

One of the fundamental problems lying in the range between graph theory and computer science is to compute a minimum spanning tree (MST) of an undirected weighted graph, i.e., an acyclic connected subgraph on all vertices which has minimum weight. The standard (network design) application for computing a minimum spanning tree could be as follows: Assume that there are several offices and we want to lease phone lines to connect them with each other. The phone company charges different amounts of money to connect different pairs of offices. Now, the task is to find a set of lines that connects all offices with a minimum total cost. This problem is equivalent to the computation of a minimum spanning tree in the undirected weighted graph G which is constructed as follows: The offices are the vertices of G, a line between two offices corresponds to an edge in G, and the weight function maps each edge to the amount of money which has to be payed for the corresponding phone line.

The history of minimum spanning tree algorithms dates back to Borůvka [Bor26] and Jarník [Jar30]. Today's most popular textbook algorithms, which are presented in nearly each course dealing with efficient algorithms or theoretical computer science, are those by Kruskal [Kru56] and Prim [Pri57]. A survey including minimum spanning tree algorithms up to 1985 is given by Graham and Hell [GH85]. Although there exists a minimum spanning tree algorithm by Pettie and Ramachandran [PR02], which is known to have an optimal runtime, the actual function of the running time is still unknown. Currently, the best known upper bound on the runtime is $\mathcal{O}(m\alpha(m,n))$ (see also [Pet99; Cha00]), where $\alpha(m,n)$ is the inverse of the Ackermann function. A randomized algorithm computing a minimum spanning tree in linear time $\mathcal{O}(m)$ with high probability was presented by Karger, Klein, and Tarjan [KKT95]. This algorithm makes use of the discovery that given a spanning tree T, it can be verified in linear time whether or not T is a minimum spanning tree (see [Kom85; DRT92; Kin97]). In addition to a variety of sequential algorithms, there also exist several parallel as well as distributed algorithms [GHS83; JM95; CC96; LPSPP05; Elk06].

Several algorithms perform a *greedy strategy*. They start with an empty set of edges. Subsequently, they choose an edge of minimum weight satisfying certain properties. Two principles to construct a minimum spanning tree were described by Tarjan [Tar83], which he called the *red rule* and the *blue rule*, respectively. The red rule states that an unique heaviest edge e of a cycle C does not belong to any minimum spanning tree. Furthermore, if e is a (not necessarily unique) heaviest edge of C, there exists a minimum spanning tree that does not contain the edge e. The blue rule states that an unique lightest edge in some cutset belongs to every minimum spanning tree. Whereas a lightest edge of some cutset must belong to some MST.

The algorithms by Kruskal and Prim make use of the blue rule. Applying these algorithms, it is possible to obtain different minimum spanning trees (provided that there is more than one MST). In this case, the computed MST depends on the order in which the edges are considered. Moreover, if there exist different minimum spanning trees then it is possible to transform any of these MSTs T into another MST by performing exactly one edge swap. In particular, it is possible to transform T into any other MST T' by performing several edge swaps *without* leaving the class of minimum spanning trees, i.e., each spanning tree T'' that is obtained along this way from T to T' has minimum weight. This property was observed by Ford and Fulkerson [FF62]. Further developments and generalizations [KKS78; Kan87; MP92] regarding the so-called tree graph are considered in this thesis.

Weighted Spanning Trees and the Tree Graph

The *tree graph* of an undirected connected graph G is defined in the context of edge swaps between spanning trees. The vertex set of G's tree graph is the set of all spanning trees in G. Two spanning trees are joined by an edge if and only if they are related by a single edge swap. There are various questions that arise regarding tree graphs, some of which were discussed by Kano [Kan87]. He proposed four conjectures concerning distances (in regard to the number of edge swaps) between spanning trees of different weights in the tree graph [Kan87]. His work was mainly motivated by a paper by Kawamoto, Kajitani, and Shinoda [KKS78]. For example, one of his conjectures was that a kth smallest spanning tree (a so-called k-MST) can be obtained by performing at most $k-1$ edge swaps starting with an arbitrary minimum spanning tree. Note that the opposite direction is easy, i.e., a minimum spanning tree can be obtained by performing at most $k-1$ (weight-reducing) edge swaps starting with an arbitrary k-MST. Actually, this conjecture of Kano was proven by Mayr and Plaxton [MP92] resulting in an algorithm with runtime $\mathcal{O}((mn)^{k-1})$ for the ($\mathcal{NP}$-hard) KMST problem, i.e., the problem of determining whether or not a given graph has k spanning trees with distinct weights less than or equal to a threshold B. Note that the number of different weighted spanning trees can be counted by using an extension of the well-known matrix tree theorem (due to Kirchhoff) [BM97], which consists of evaluating the determinant of the so-called Laplacian matrix. An application of the KMST problem could be as follows. In many real-world problems, there exists more than one objective function when solving a network problem like computing a spanning tree. For example, we can measure different kinds of costs or benefits. In general, there does not have to be a spanning tree which is optimal with respect to each of these objective functions, i.e., we have to find a trade-off between numerous possibilities. Regarding different spanning trees, the question is how to measure or compare their performance. On the one hand, it is possible to measure the performance of a spanning tree by the absolute or relative values of the objective functions. On the other hand, it is also reasonable to ask for the rank of the spanning tree regarding a certain function. For example, if a spanning tree is a k-MST for a large value of k then there is a large room of improvement whereas a second smallest spanning tree can hardly be improved.

In addition to the proof of one of Kano's conjectures, Mayr and Plaxton formulated a new conjecture which unifies Kano's remaining three conjectures. This thesis addresses this unified conjecture. More precisely, we make some progress in proving the following claim: Let $B = (V, P, Q, w)$ be a weighted bispanning graph (a graph whose edge set E consists of two edge-disjoint spanning trees) such that $w(P) < w(Q)$ and Q is the only spanning tree with weight $w(Q)$ in B. Then, it is conjectured that there are at least $|V|-1$ spanning trees in B which have pairwise different weights strictly less than $w(Q)$. In this thesis, we show that this is true if the spanning tree P is the only spanning tree with weight $w(P)$. Furthermore, we prove that there are sufficiently many distinct spanning trees if the given bispanning graph has no minor isomorphic to the complete graph on four vertices (K_4). In this context, we analyze the spanning tree structure of a graph using some findings from matroid theory. Based on these ideas, we present a slightly refined conjecture, namely that it might be sufficient to count only so-called partition spanning trees. We support this theory with several theorems and by the identification of a subclass of bispanning graphs to which the problem can be reduced. The graphs of this class have the property of containing a minor isomorphic to the K_4. For the K_4 itself, we are able to prove our conjecture through a tedious case analysis.

Base Orderings

Strongly related to the above problem of counting weighted spanning trees are so-called base orderings, which have their origin in matroid theory. Each path between two spanning trees, T and T', in the tree graph corresponds to a sequence of edge swaps transforming each of these spanning trees into the other. Assuming that T and T' are disjoint, each edge of T is eventually exchanged with an edge of T'. Hence, we can order the edges of T and T' according to the step in which the edge is exchanged in the corresponding edge swap sequence. For this reason, we associate with each path in the tree graph a so-called *base ordering*. Here, the term 'base' has its origin in matroid theory, where bases are the maximum independent subsets of a certain ground set [Oxl92; Tut71]. Then, the spanning trees of a graph G are the bases of the well-known cycle matroid of G. Note that properties of the cycle matroid of a graph cannot easily be extended to general matroids since the cycle matroid of a graph fulfills additional properties. Thus, we consider 'spanning tree orderings'. As a result, we present a new kind of base ordering which is a subset of the known cyclic base orderings [Wie06]. We will call them *subsequence-interchangeable base orderings*. Regarding two spanning trees T and T' of the tree graph, their meaning is as follows. A subsequence-interchangeable base ordering of T and T' corresponds to a path (consisting of single edge swaps) such that each (consecutive) subsequence of edge swaps of this path corresponds to a path connecting T with another spanning tree T'' in the tree graph. We study these orderings and discuss several approaches to constructing them. Although we cannot prove their existence for all possible pairs of spanning trees, we have not found a counterexample, yet. If we succeed in a proof, it might be possible to use them for proving Mayr and Plaxton's conjecture: Let $B = (V, P, Q, w)$ be a weighted bispanning graph such that Q is the only spanning tree with weight $w(Q)$ and let $\mathcal{P}$ be a path between Q and P which corresponds to a subsequence-interchangeable base ordering. Then, the spanning trees on $\mathcal{P}$ have pairwise different weights.

Approximation of Spanning Trees

Furthermore, we consider a problem, which is related to the simplification of graphs with respect to the number of edges. Problems of this kind can be summarized by the term *network sparsification*. The intention is to thin out the graph while retaining certain network characteristics, e.g., the distances between node pairs or the centrality measures of the nodes. The aim of this task is to reduce the complexity of a given graph in order to simplify computations of network problems or to feature a concise visualization of a complex network with its most important structural properties. For example, the network can be made more amenable to visual examination.

Carrying this sparsification to an extreme, we would require the resulting graph to be a spanning tree, since the elements of this graph class have a minimum number of edges among all connected subgraphs and they offer a variety of beneficial properties which can be exploited for fast network algorithms even if the considered problems are (in general) $\mathcal{NP}$-hard.

In this thesis, we analyze the problem of computing a spanning tree of a graph, that minimizes, in its simplest form, the sum of the distances between all pairs of nodes, that were connected by an edge in the original graph. Actually, we consider a more general form, where the sum is computed of pth powers of the respective distances (or distance differences), i.e., the calculation is made with respect to the L_p-norm.

The problem is related to a couple of other problems. A simliar problem is computing distance-minimizing or distance-approximating spanning trees (DMST, DAST, [EKM+08]). In contrast to the setting in this thesis, the DMST and DAST problems consider the distances of *all* vertex pairs (instead of only pairs connected by single edges in the original graph). Both problems, DMST and DAST, were shown to be $\mathcal{NP}$-complete for all norms L_p, $p \in \mathbb{N}$. For both problems a fixed-edges variant was introduced in [EKM+08], where the input includes a set of fixed edges that have to appear in any admissible solution. For this fixed-edges version of DAST and arbitrary L_p-norms, there is no constant-factor approximation unless $\mathcal{P} = \mathcal{NP}$. The simplest case of the DMST problem, i.e., DMST using the L_1-norm, is equal to the Simple Network Design Problem introduced in [JLRK78] as well as the problem of computing a Minimum Average Distance (MAD) Tree [DDGS03]. Moreover, this problem is equivalent to the DAST problem with respect to the L_1-norm. In the more general form of the Network Design Problem, we are given a weighted undirected graph and want to compute a connected subgraph that respects a certain budget constraint (regarding the sum of the edge weights) and that minimizes the sum of all shortest path lengths. Of course, this problem was also shown to be $\mathcal{NP}$-complete [JLRK78].

A similar relationship exists between GAST and the problem of computing a minimum fundamental cycle basis (MIN-FCB). Again, we are given a weighted undirected graph. The aim is to compute a spanning tree (or the respective cycle basis), that leads to a minimum sum of weights of all fundamental cycles (induced by the edges of the spanning tree). Deo et al. [DPK82] have shown $\mathcal{NP}$-completeness of this problem. Galbiati and Amaldi [GA04] proposed an $2^{\mathcal{O}(\sqrt{\log n \log\log n})}$-approximation algorithm for arbitrary graphs. Their approach used a related problem introduced by Hu, namely the Minimum Communication Cost Spanning Tree Problem (MCT) [Hu74], which was shown to be approximable within the same factor by Peleg and Reshef [PR98]. The currently best known approximation ratio is due to Elkin, Emek, Spielman, and Teng who presented an approximation algorithm with ratio $\mathcal{O}(\log^2 n \log\log n)$.

1.2 Outline

The remaining chapters of this thesis are organized as follows. In Chapter 2, we introduce several definitions of graph theory. In particular, we analyze so-called bispannable and bispanning graphs. We also present operations to construct and modify these graphs. Thereafter, we discuss possibilities to compute two edge-disjoint spanning trees with an application for the Shannon switching game. In Chapter 3, we present Kano's as well as Mayr and Plaxton's Conjectures concerning distances of spanning trees that have different weights. Furthermore, we analyze bispanning graphs with special weight functions and a special structure. Here, we give a brief introduction to matroid theory, which we use for this analysis. Afterward, we propose a slightly refined conjecture in Chapter 4. This conjecture states that it is sufficient to count only spanning trees which define a new partition of a bispannable graph. The analysis of bispanning graphs with a special weight function and a special structure as well as the refined conjecture were presented at the 19th International Workshop on Combinatorial Algorithms 2008 (IWOCA'08) [Bau08]. Subsequence-interchangeable base orderings are introduced in Chapter 5. In Chapter 6, we consider the problem GAST with respect to the L_p-norm and we show for $p = 1$ a reduction to MIN-FCB. The proof of $\mathcal{NP}$-completeness was published as a technical report [BT08].

Chapter 2

Preliminaries

2.1 Graphs, Trees, and Tree Graphs

Throughout this thesis, an *undirected (multi) graph* $G = (V, E)$ is a pair consisting of a finite set V and a (multi) subset E of all 2-elementary subsets of V. The set V is called *vertex set* and E is called *edge set* of G. Unless stated otherwise, we denote the number of vertices by $n := |V|$ and the number of edges by $m := |E|$. A (multi) graph is called *simple* if it contains no multiple edges. An *undirected weighted graph* $G = (V, E, w)$ is given by an undirected graph $G = (V, E)$ and a *weight function* $w\colon E \to \mathbb{R}$ associating a weight $w(e)$ with each edge $e \in E$. For any subset $E' \subseteq E$, we define the weight of E', denoted by $w(E')$, as the sum of the weights of all edges in E', that is,

$$w(E') := \sum_{e \in E'} w(e) \ .$$

A graph $G' = (V', E')$ is called a *subgraph* of an undirected graph $G = (V, E)$ if $V' \subseteq V$ and E' is a (multi) set with

$$E' \subseteq \big\{ e \in E \mid e = \{v, w\} \text{ with } v, w \in V' \big\} \ . \tag{2.1}$$

We call G' an *induced subgraph* if (2.1) holds with equality. In this case, we say that G' is induced by V'. If $|V'| > 1$ and $V' \neq V$ holds, we call G' a *non-trivial* (induced) subgraph. Given a graph $G = (V, E)$ and a vertex set $V' \subseteq V$, we denote by $G[V']$ the subgraph of G induced by V' and we define by $G \setminus V'$ the graph induced by $V \setminus V'$. For any edge $e \in E$, we denote by $G \setminus e$ the graph $G' = (V, E \setminus \{e\})$. Analogously, for any subset $E' \subseteq E$, we define $G \setminus E'$ to be the graph $G' = (V, E \setminus E')$. Moreover, we denote by G/e the graph we obtain by contracting an edge $e = \{v, w\} \in E$. Contracting e means that we remove e and identify v and w, i.e., we merge v and w to a new vertex. More formally, contracting an edge $e = \{v, w\}$ in $G = (V, E)$ yields the graph $G' = (V', E')$ with vertex set $V' = (V \setminus \{v, w\}) \cup \{v_e\}$ where v_e is a new vertex. The edge set E' is defined to be the multiset (i.e., the contraction can generate multiple edges) with

$$\begin{aligned} E' = \big\{ &\{x, y\} \in E \mid \{v, w\} \cap \{x, y\} = \emptyset \big\} \\ &\uplus \big\{ \{v_e, z\} \mid \text{for each } \{x, z\} \in E \text{ with } x \in \{v, w\} \text{ and } z \in V \setminus \{v, w\} \big\} \ . \end{aligned}$$

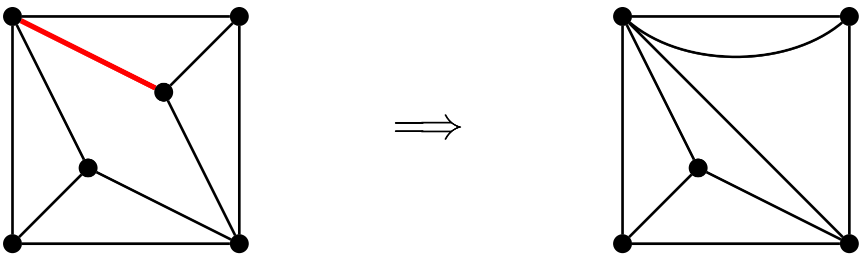


Figure 2.1: Contracting an edge.

For any subset $E' \subseteq E$, we define G/E' to be the graph we obtain by contracting all edges in E' (in any order). Analogously, it is also possible to define the contraction G/V' for any vertex set $V' \subseteq V$: we identify all vertices in V' by a new vertex v', remove all edges joining vertices in V', and redefine any edge $\{v, w\}$ with $v \in V'$ and $w \notin V'$ to $\{v', w\}$. In the context of contracting and deleting edges, a so-called minor of a graph can be defined. More precisely, a graph H is a *minor* of a graph G if H can be obtained from G by a series of deletions and contractions of edges and by deletions of vertices. In Figure 2.2, we show that the complete graph on four vertices K_4 is a minor of the left graph: we only have to remove the blue edges and the blue vertex, and contract the red edge. A graph H is a *topological minor* of a graph G if there exists a subgraph G' of G which can be obtained from H by several subdivisions of H's edges. An edge-subdivision is an operation which replaces an edge $= \{v, w\}$ by a sequence of edges $e_1, \ldots, e_k$ with $e_1 = \{v, v_1\}, e_2 = \{v_1, v_2\}, \ldots, e_k = \{v_{k-1}, w\}$ such that $v_1, \ldots, v_{k-1}$ are new vertices. Note that each topological minor of a graph is also a minor whereas a minor does not need to be a topological minor. The K_4 is a topological minor as well as a minor of the left graph in Figure 2.2. The property of a graph to have a minor

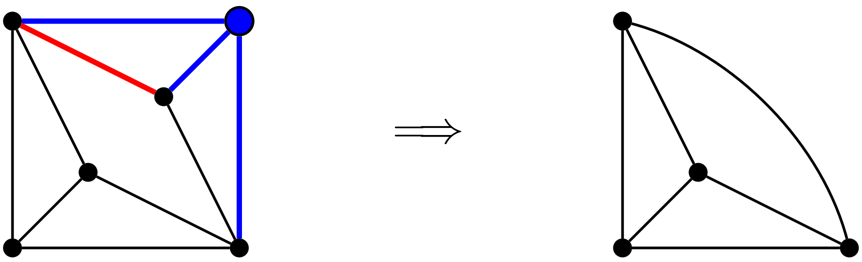


Figure 2.2: A minor of a graph.

isomorphic to the K_4 plays a central role throughout this thesis (cf. Proposition 4.9). Two graphs $G = (V, E)$ and $H = (W, F)$ are said to be *isomorphic* if there exists a bijection $\varphi \colon V \to W$ such that $\{v, w\} \in E$ holds if and only if $\{\varphi(v), \varphi(w)\} \in F$ for all $v, w \in V$. Such a map φ is called an *isomorphism*.

Let $G = (V, E)$ be an undirected graph. Two vertices $v, w \in V$ are called *adjacent* if there exists an edge $e \in E$ with $e = \{v, w\}$. For an edge $e = \{v, w\} \in E$, we say e

is *incident* to v and w and vice versa. The number of incident edges of a vertex v is called the *degree* of v and is denoted by $\deg(v)$. Given two vertices $v, w \in V$, an *undirected path* between v and w is a sequence $P = (v_0, \ldots, v_k)$ of distinct vertices such that $v = v_0$, $w = v_k$, and $\{v_i, v_{i+1}\} \in E$ for all $i \in \{0, \ldots, k-1\}$. The *length of an undirected path* P is defined as the number of edges associated with P. A (simple) *cycle* is defined like an undirected path with the difference that v equals w (but all other vertices are distinct) and each edge is used at most once. Hence, the shortest cycle is a pair of multiple edges. An undirected graph $G = (V, E)$ is said to be *connected* if there exists an undirected path between v and w for all $v, w \in V$. An edge $e \in E$ is called *bridge* if $G \setminus e$ is not connected. We call a vertex $v \in V$ a *cut vertex* if the removal of v (and its incident edges) will disconnect the graph. A graph $G = (V, E)$ is called *k-vertex-connected* if $G \setminus V'$ is connected for any vertex set $V' \subseteq V$ of size $|V'| < k$. For the sake of completeness, if G is completely connected (i.e., every two vertices are joined by an edge), we define G to be $(|V| - 1)$-connected. The largest integer value k such that G is k-vertex-connected is called the *vertex-connectivity* of G which is denoted by $\kappa(G)$. The graph G is said to be *ℓ-edge-connected* if $G \setminus E'$ is connected for any edge set $E' \subseteq E$ of size $|E'| < \ell$. The *edge-connectivity* $\lambda(G)$ is the largest integer value ℓ such that G is ℓ-edge-connected.

Given an undirected connected graph $G = (V, E)$, a *spanning tree* T of G is any subset of E for which the graph $G' = (V, T)$ is connected and does not contain any cycle. The set of all spanning trees of G is denoted by $\mathcal{T}(G)$. For any spanning tree $T \in \mathcal{T}(G)$ and an edge $f \in E \setminus T$, we denote by $C(T, f)$ the unique cycle (the so-called *fundamental cycle*) of G defined by f with respect to T. Given a pair of distinct edges $e, f \in E$ such that $e \in C(T, f)$, we define the ordered pair (e, f) to be a single *edge swap* where the edge e is called *leaving edge*, and f is called *entering edge* . Thus, the set $T' = (T \setminus \{e\}) \cup \{f\}$ is a spanning tree, too. We say that T and T' are *related by an edge swap.* Clearly, the ordered pair $(f, e) := (e, f)^{-1}$ is an edge swap of T'.

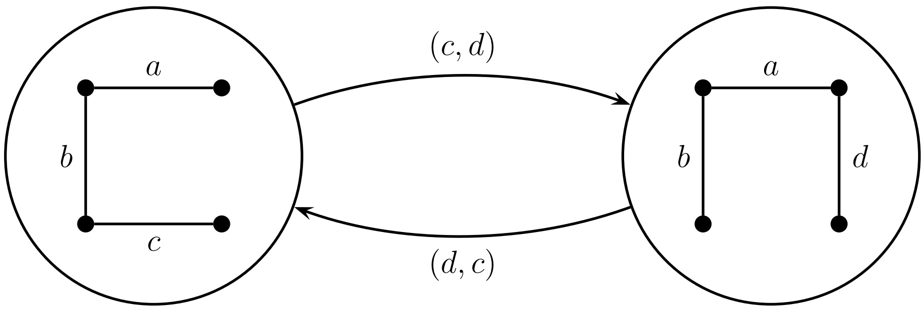
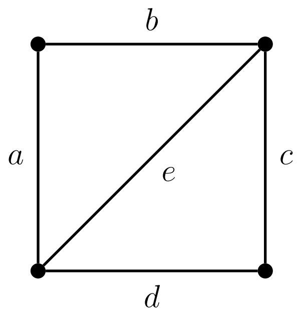
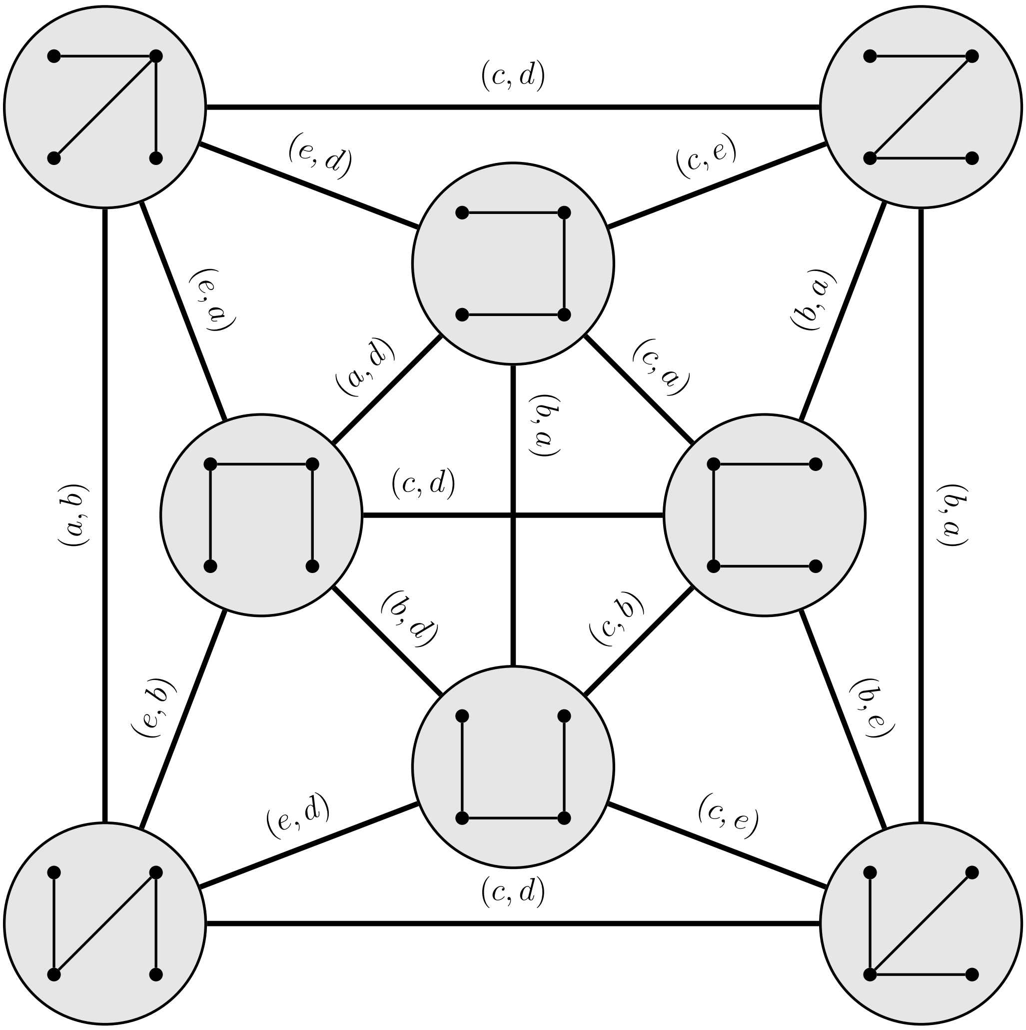


Figure 2.3: Two spanning trees related by an edge swap.

The *tree graph* of an undirected connected graph $G = (V, E)$ is the graph $\mathcal{G}(G) = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \mathcal{T}(G)$ and there exists an edge $\{T, T'\} \in \mathcal{E}$ between two spanning trees $T, T' \in \mathcal{V}$ if and only if T and T' are related by an edge swap. In Figure 2.4, there is an illustration of an undirected graph and its tree graph.



(a) An undirected graph with four vertices and five edges.



(b) The tree graph of (a) and its edge swaps.

Figure 2.4: A graph and its tree graph

2.2 Bispannable and Bispanning Graphs

The focus of this thesis is directed to the analysis of so-called bispanning graphs, which are introduced in this section. To this end, we need the following definition. Let $S = \{s_1, s_2 \ldots, s_k\}$ be a set, then a *partition* of S is a set of non-empty disjoint subsets of S such that each element of S is in exactly one of these subsets. We call a partition $\mathcal{P}$ of S *non-trivial* if $\mathcal{P} \neq \{S\}$ and $\mathcal{P} \neq \{\{s_1\}, \{s_2\} \ldots, \{s_k\}\}$ holds.

Definition 2.1. *A connected, undirected graph $G = (V, E)$ is called a bispannable graph if there exists a partition of E into two spanning trees. Given a partition of E into two spanning trees P and Q, the triple $B = (V, P, Q)$ is said to be a bispanning graph.*

Thus, by Definition 2.1, each bispannable graph is (in general) associated with different bispanning graphs. Figure 2.5 illustrates a bispannable graph and two different partitions of its edge set into two spanning trees which are emphasized by different colors. We remark that in each picture of this thesis related to bispanning graphs, the spanning tree edges of P are always colored in red whereas the spanning tree Q is always blue-colored. Although it is possible to switch between P and Q, there are some properties concerning only one of them. Note that both bispanning graphs in Figure 2.5 are different (in terms of not being isomorphic) since the right graph consists of a blue and a red path whereas the other graph consists only of a red path (the blue spanning tree is not a path).

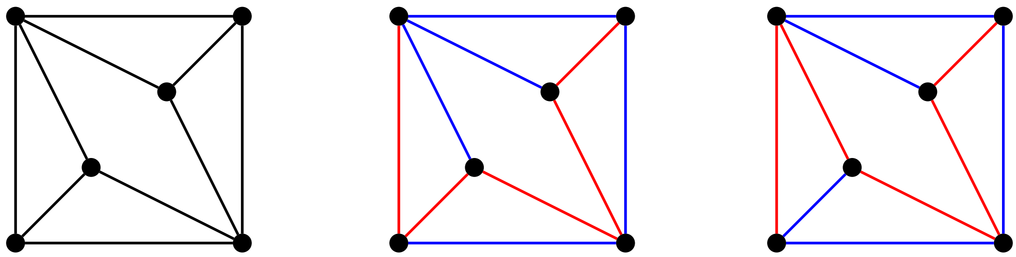


Figure 2.5: A bispannable graph and two different partitions into bispanning graphs.

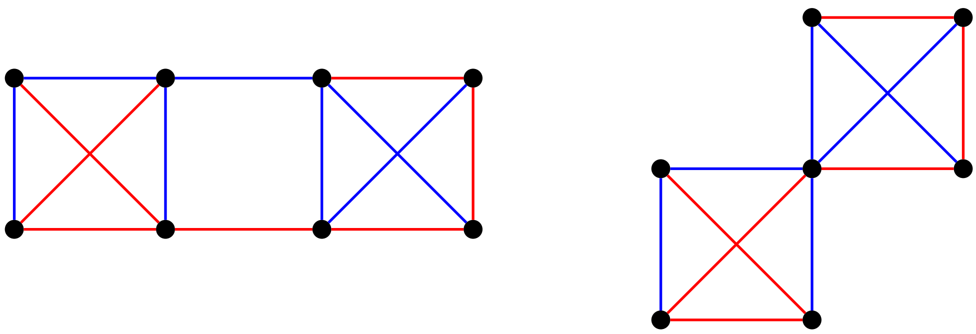
In the following, we want to explore some fundamental properties of bispanning graphs. Let $G = (V, E)$ be an arbitrary bispannable graph on n vertices. First of all, we note that G contains exactly $m = 2n - 2$ edges. Thus, the sum of all degrees is exactly

$$\sum_{v \in V} \deg(v) = 2m = 4n - 4 \,. \tag{2.2}$$

Together with the observation that each vertex $v \in V$ has degree $\deg(v) \geq 2$, we obtain the following lemma by applying the pigeon-hole principle.

Lemma 2.2. *Each bispannable graph $G = (V, E)$ with $|V| > 1$ contains a vertex v of degree $\deg(v) = 2$ or degree $\deg(v) = 3$.*

Moreover, if a bispannable graph contains no vertex v of degree $\deg(v) = 2$, there have to be four vertices of degree three. A related property is the vertex- and edge-connectivity. By Definition 2.1, any bispannable graph $G = (V, E)$ is 1-vertex-connected and 2-edge-connected since there are at least two edge-disjoint paths between any two vertices v and w according to any partition of E into two spanning trees. Moreover, a bispannable graph G does not need to be 2-vertex-connected or 3-edge-connected (see Figure 2.6) but G can be up to 3-vertex and 3-edge-connected (e.g., the wheels which are considered in Chapter 5). An edge-connectivity (and vertex-connectivity, respectively) greater than 3 is impossible since each bispannable graph contains a vertex v of degree $\deg(v) \leq 3$. A summary of all possible combinations is presented in Table 2.1 (clearly, we have $\kappa(G) \leq \lambda(G)$ for each graph G).



(a) A 2-edge-connected bispanning graph. (b) A 1-vertex-connected bispanning graph.

Figure 2.6: Connectivity of bispanning graphs.

		edge-connectivity λ		
		2	3	≥ 4
vertex-connectivity κ	1	+	+	−
	2	+	+	−
	3		+	−
	≥ 4			−

Table 2.1: Vertex- and edge-connectivity of bispanning graphs.

We have seen that each bispannable graph has to fulfill certain connectivity properties. These properties are necessary but not sufficient since for example a 2-vertex-connected and 3-edge-connected graph does not need to contain two edge-disjoint spanning trees. A more precise property of an undirected graph $G = (V, E)$ implying the existence of k edge-disjoint spanning trees was proven by Nash-Williams [NW61; NW64]. The same result was investigated independently by Tutte [Tut61].

Proposition 2.3. *A graph $G = (V, E)$ has k edge-disjoint spanning trees if and only if*

$$|E(\mathcal{P})| \geq k \cdot (|\mathcal{P}| - 1) \tag{2.3}$$

for each partition $\mathcal{P}$ of V. We denote by $E(\mathcal{P})$ the set of edges in E which join vertices belonging to different members of $\mathcal{P}$.

Some consequences of this proposition concerning the connectivity of an undirected graph were discussed by Gusfield [Gus83]. We remark that the original proof by Nash-Williams is quite intricate. A shorter version using several technical findings of matroid theory can be found in [Wes00]. Proposition 2.3 implies that a graph $G = (V, E)$ is bispannable if and only if it contains $2n - 2$ edges and (2.3) is fulfilled with $k = 2$ for each partition $\mathcal{P}$ of V. Unfortunately, Equation (2.3) is not useful for an algorithm to detect whether a graph contains two edge-disjoint spanning trees or not.

As aforementioned, the main focus of this thesis lies on the analysis of bispanning (or bispannable) graphs. A question which turns out to be very important while studying these graphs is whether they are composite or not. The formal definition of this property is as follows.

Definition 2.4. *A bispannable graph $G = (V, E)$ is called* atomic *if G contains no non-trivial bispannable subgraph. Otherwise, G is called* composite*.*

This definition can be extended to bispanning graphs. Then, a bispanning graph $B = (V, P, Q)$ is composite (atomic) if the underlying bispannable graph is composite (atomic). Both bispannable graphs in Figure 2.6 are composite since each of them consists of two non-trivial bispanning subgraphs. The latter observation is obvious once we have detected a cut vertex or a cut containing only two edges. An example of an atomic bispannable graph is given in Figure 2.5. The following theorem proposes a characteristic of atomic bispannable graphs by extending Proposition 2.3.

Theorem 2.5. *A bispannable graph is atomic if and only if (2.3) with $k = 2$ is a strict inequality for each non-trivial partition $\mathcal{P}$ of V.*

Proof. To establish the 'only if' direction, we will prove the contrapositive, i.e., we assume (2.3) is an equality for some non-trivial partition and show that the given graph must be composite. To this end, we suppose there exists a non-trivial partition $\mathcal{P} = \{V_1, V_2, \ldots, V_\ell\}$ of V such that $|E(\mathcal{P})| = 2\ell - 2$. Let $\ell_i := |V_i|$ for $i = 1, \ldots, \ell$. We observe that there are at most $2\ell_i - 2$ edges joining vertices of component V_i for each $i = 1, \ldots, \ell$. Hence, the number of all these edges is at most

$$\sum_{i=1}^{\ell} (2\ell_i - 2) = 2n - 2\ell \ .$$

Moreover, there are exactly $2(\ell - 1)$ edges which join vertices belonging to different members of $\mathcal{P}$. Thus, the number of remaining edges is

$$(2n - 2) - (2\ell - 2) = 2n - 2\ell \ .$$

Therefore, each component $G[V_i]$ consists of $2\ell_i - 2$ edges. Since $\mathcal{P}$ is a non-trivial partition of V, there exists an index $i' \in \{1, \ldots, \ell\}$ such that $\ell_{i'} > 1$. Then, $G[V_{i'}]$ is a non-trivial bispannable subgraph of G contradicting G to be atomic.

The 'if' direction is proven analogously. We assume the given bispannable graph is composite and show that (2.3) is an equality for some partition. To this end, we suppose the bispannable graph $G = (V, E)$ is composite and construct a partition of the vertex set such that (2.3) is an equality. To this end, let V' be any subset of V such that $G[V']$ is a non-trivial bispannable subgraph of G. If V' contains $\ell := |V'|$ vertices then there are $2\ell - 2$ edges in G joining vertices of V'. Let $V'' := V \setminus V' = \{v_1, \ldots, v_j\}$ the set of all vertices which are not contained in V'. Clearly, it holds $j = n - \ell \geq 1$ since $G[V']$ is a non-trivial subgraph. Now, we consider the partition

$$\mathcal{P} = \big\{\{v_1\}, \ldots, \{v_j\}, V'\big\} .$$

Then, the number of edges joining different members of $\mathcal{P}$ is exactly

$$(2n - 2) - (2\ell - 2) = 2j .$$

Hence, $\mathcal{P}$ is a partition such that $|E(\mathcal{P})| = 2 \cdot (|\mathcal{P}| - 1)$ proving the theorem. □

2.2.1 Construction of Bispanning Graphs

In the previous section, we have discussed several properties which a graph $G = (V, E)$ has to meet in order to be an (atomic) bispannable graph. Now, we consider a method to construct an arbitrary (not necessarily atomic) bispannable graph. To this end, we define two different operations which are called *double-leaf attachment* and *edge-split*. Both operations can be applied to any bispannable graph $G = (V, E)$ in order to preserve the resulting graph being bispannable, too. Hence, we can start with any bispannable graph $G = (V, E)$ which is possibly degenerated to a single vertex. The operations to modify G are defined as follows.

(i) *Double-leaf attachment* of vertex u and v

Let $u, v \in V$ be two (not necessarily distinct) vertices in G. We introduce a new vertex w and connect it by two edges, $\{u, w\}$ and $\{v, w\}$, with G.

The resulting graph $G' = (V', E')$ with vertex set $V' = V \cup \{w\}$ and edge set $E' = E \cup \{\{u, w\}, \{v, w\}\}$ is a bispannable graph since G can be partitioned into two edge-disjoint spanning trees, P and Q, and the new edges can be regarded as leaves which are appended to P and Q, respectively. For this reason, the operation is called 'double-leaf attachment'.

Note that if u and v are not distinct, i.e., both variables stand for the same vertex, we introduce a pair of multiple edges. The operation double-leaf attachment is illustrated in Figure 2.7.

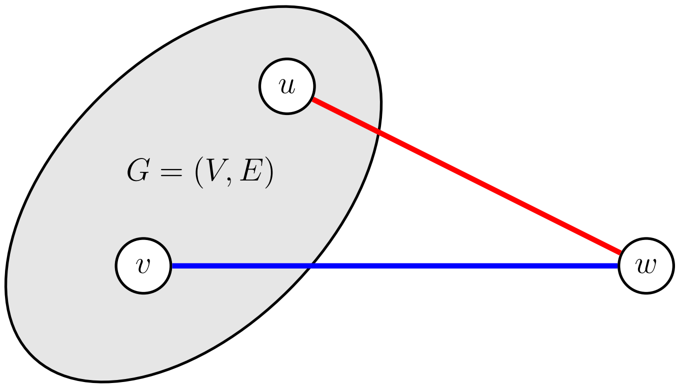


Figure 2.7: Operation double-leaf attachment.

(ii) *Edge-split* of edge e by vertex v

Let $v \in V$ be a vertex and $e = \{x, y\} \in E$ be an edge in G. We split the edge e into two edges, that is, we introduce a new vertex w and connect it with both vertices of e followed by the removal of e. Afterwards, the vertices v and w are joined by an edge.

The resulting graph $G' = (V', E')$ with vertex set $V' = V \cup \{w\}$ and edge set $E' = (E \setminus \{e\}) \cup \{\{x, w\}, \{w, y\}, \{v, w\}\}$ is again a bispannable graph: Since G is bispannable, it can be partitioned into two edge-disjoint spanning trees P and Q. Without loss of generality, we assume $e \in P$. Then, $P' = (P \setminus \{e\}) \cup \{\{x, w\}, \{w, y\}\}$ and $Q' = Q \cup \{v, w\}$ is an admissible partition of G' into two disjoint spanning trees. In Figure 2.8, the dotted edge $e \in E$ is splitted with a blue edge into two parts.

Remark: According to Figure 2.8, we often say that the edge e is split by the edge $\{v, w\}$ (into $\{x, w\}$ and $\{y, w\}$). Then, $\{v, w\}$ is called the *splitting edge*.

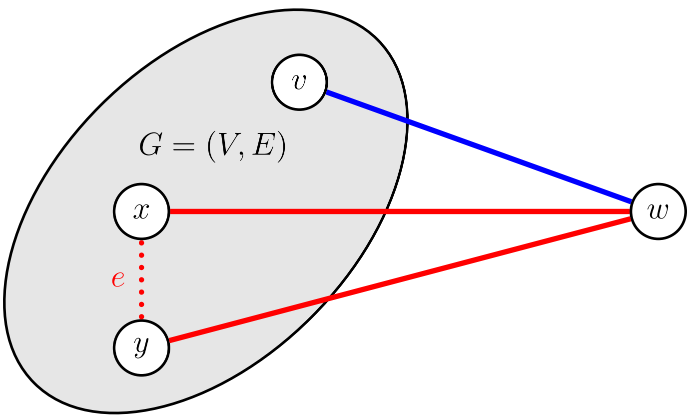


Figure 2.8: Operation edge-split.

As mentioned above, it is possible to extend a bispannable graph using the operations given above. In the following, we will see that applying these operations to a single vertex suffices in order to generate each bispannable graph $G = (V, E)$.

Theorem 2.6. *Each bispannable graph $G = (V, E)$ can be constructed by a sequence of 'double-leaf attachment' and 'edge-split' operations starting with a single vertex.*

Proof. We prove this theorem by induction over the number $n := |V|$ of vertices of a given bispannable graph $G = (V, E)$. If G consists of $n = 2$ vertices then G contains a pair of multiple edges. This graph can be constructed by applying a double-leaf attachment operation to a single vertex.

Hence, we suppose that $G = (V, E)$ is a bispannable graph with $n > 2$ vertices. By Lemma 2.2, the graph G contains a vertex v of degree $\deg(v) = 2$ or $\deg(v) = 3$. Furthermore, there exists a partition of E into two spanning trees P and Q.

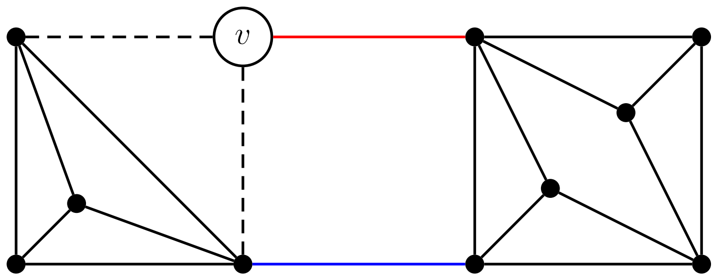
If G contains a vertex v of degree $\deg(v) = 2$ then v is connected to G by an edge $p = \{x, v\} \in P$ and an edge $q = \{y, v\} \in Q$ for some (not necessarily distinct) vertices $x, y \in V$. Then, removing v and the edges p and q yields again a bispannable graph G' since cutting off leaves of a tree preserves the property of being a tree. By induction hypothesis, there exists a sequence of 'double-leaf attachment' and 'edge-split' operations to construct G'. Now, we extend this sequence by applying a double-leaf attachment operation of vertex x and y resulting in G.

On the other hand, we suppose that G contains a vertex v adjacent to exactly three vertices v_1, v_2, and v_3. Then, either v is a leaf of P, or v is a leaf of Q, respectively. Let $\{v, v_1\}$ be the edge according to this property, and let $\{v, v_2\}$ and $\{v, v_3\}$ be the remaining two edges. We construct a bispannable graph G' as follows: Remove v and its incident edges and introduce a new edge $e = \{v_2, v_3\}$ which belongs to the same tree as the removed edges $\{v, v_2\}$ and $\{v, v_3\}$. By induction hypothesis, there exists an appropriate sequence to construct G'. Now, applying an 'edge-split' operation of e by v_1 generates the bispannable graph G. Thus, we obtain a sequence of 'double-leaf attachment' and 'edge-split' operations to construct G. This proves the theorem. $\square$

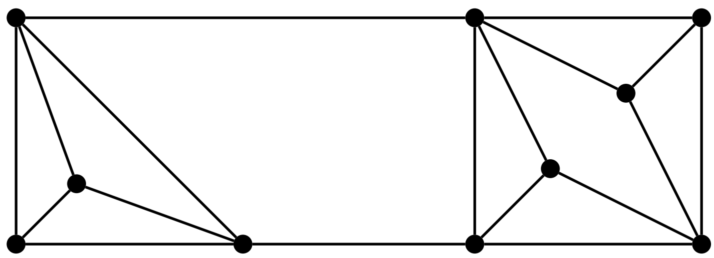
Since we do not restrict ourselves to a special partition of a bispannable graph, it is also possible to construct each bispanning graph by a sequence of 'double-leaf attachment' and 'edge-split' operations starting with a single vertex.

In the following, we discuss some remarks concerning construction sequences of bispannable/bispanning graphs. Firstly, any construction sequence for a bispannable graph $G = (V, E)$ on n vertices has length $n - 1$ since we introduce a new vertex in each step. In general, there exist many such sequences. This observation is based upon the fact that there are many possibilities to choose a vertex v of degree $\deg(v) = 2$ or $\deg(v) = 3$. A vertex v of degree $\deg(v) = 2$ can only be obtained by a double-leaf attachment since an edge-split introduces a vertex of degree 3 and increases the degree of another vertex. If there exists no vertex of degree two then (2.2) implies the existence of at least four vertices of degree 3.

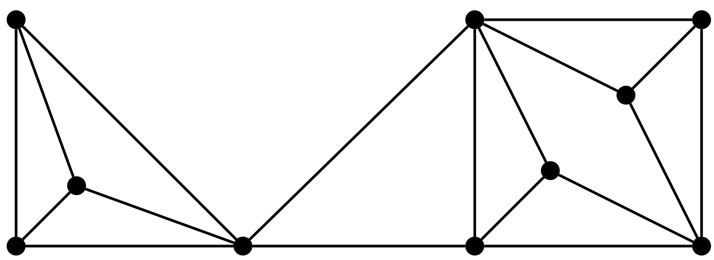
On the other hand, given a bispannable graph $G = (V, E)$ and a vertex v of degree $\deg(v) = 3$, it is not possible to choose an arbitrary edge incident to v in order to reverse an edge-split. Such a situation is illustrated in Figure 2.9. Here it is impossible that the red edge was introduced while splitting one of the dashed edges. Otherwise the blue edge becomes a bridge. In this case, the resulting graph is not bispannable any more. As a consequence, exactly one of the dashed edges in Figure 2.9 has to be colored with red.



(a) A bispanning graph.

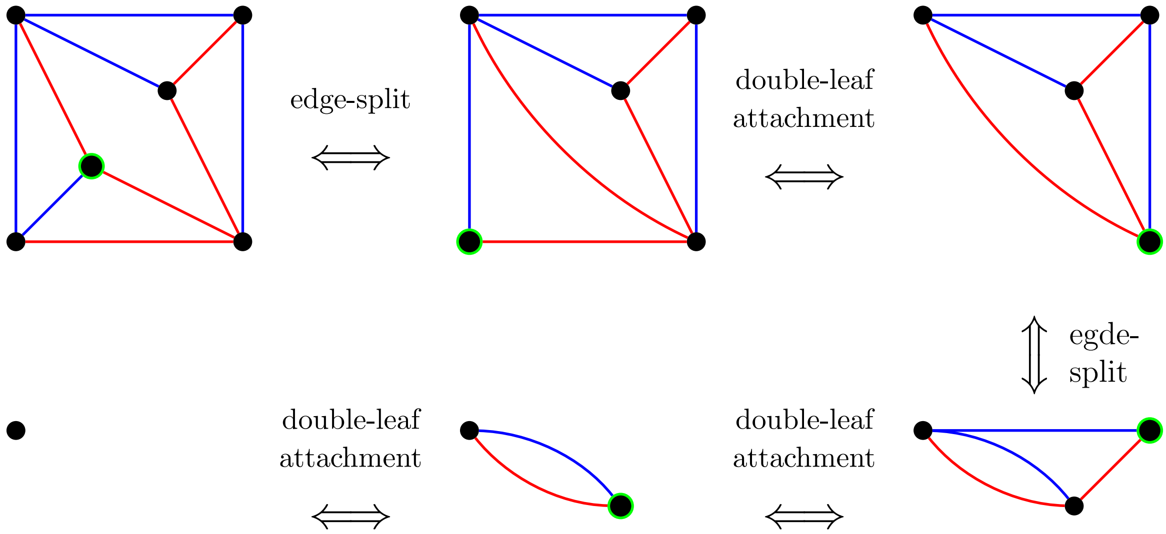


(b) Reversing the edge-split by the vertical dashed edge incident to v.

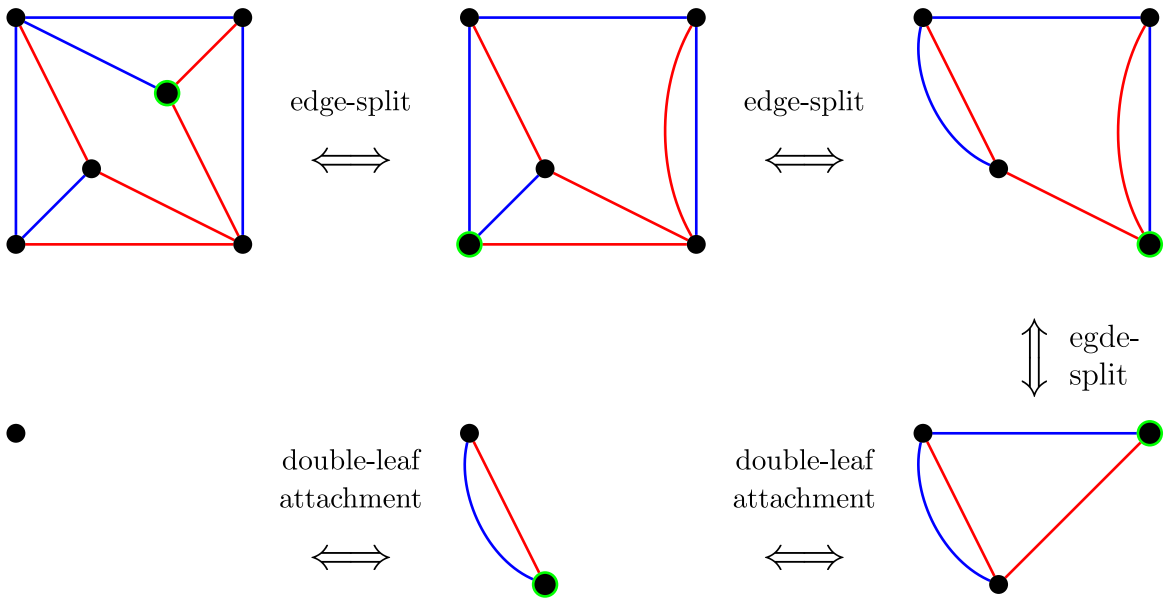


(c) Reversing the edge-split by the horizontal dashed edge incident to v.

Figure 2.9: Reversing an edge-split.



(a) First construction sequence consisting of 2 edge-split and 3 double-leaf attachment operations.

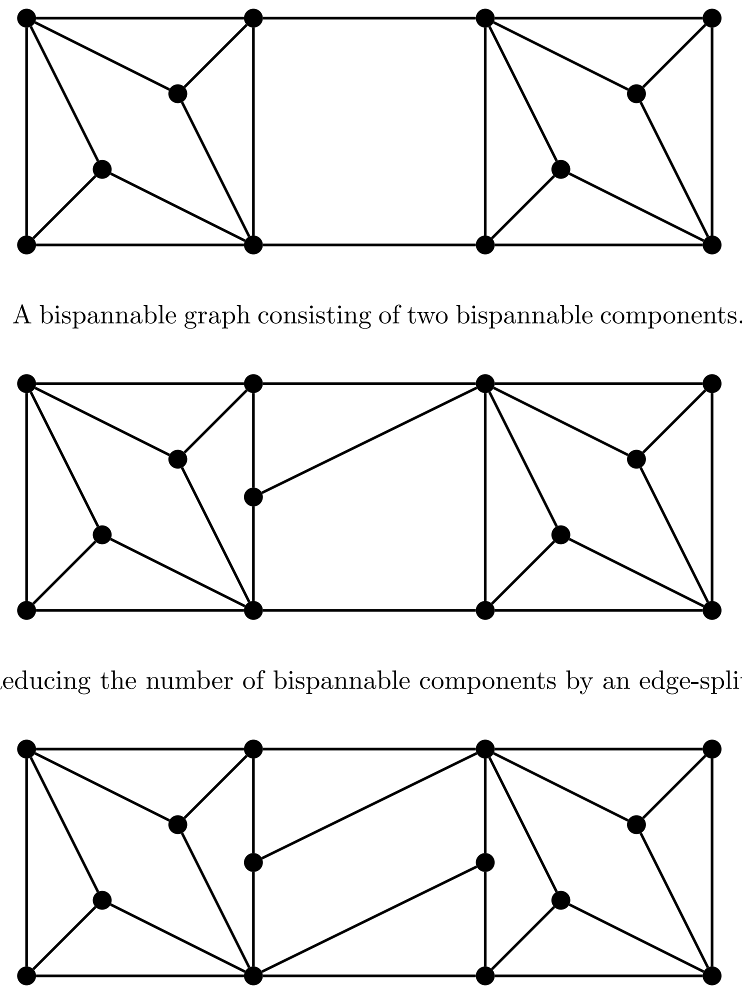


(b) Second construction sequence consisting of 3 edge-split and 2 double-leaf attachment operations.

Figure 2.10: Different construction sequences for the same graph.

In addition, it is not even required that the numbers of double-leaf attachment (edge-split) operations of different construction sequences for the same graph are equal. Figure 2.10 illustrates an example where a bispannable graph can be constructed by a sequence consisting of 2 edge-split and 3 double-leaf attachments *or* a sequence of 3 edge-splits and 2 double-leaf attachments.

Concerning the property of a bispannable graph being atomic or composite, the operations 'double-leaf attachment' and 'edge-split' have the following impact. On the one hand, each 'double-leaf attachment' applied to any atomic bispannable graph converts it into a composite one. On the other hand, it is possible to transform each composite bispannable graph into a (larger) atomic graph by using several 'edge-split' operations. In this case, the maximal number of these operations only depends on the number of disjoint bispanning components since with each edge-split at most one of them can be 'destroyed'. An illustration of this is given in Figure 2.11.



(a) A bispannable graph consisting of two bispannable components.

(b) Reducing the number of bispannable components by an edge-split I.

(c) Reducing the number of bispannable components by an edge-split II.

Figure 2.11: Using edge-splits to make a composite bispannable graph atomic.

2.2.2 Computing a Partition into two Spanning Trees

Any given bispannable graph $G = (V, E)$ can be partitioned into two edge-disjoint spanning trees P and Q yielding a bispanning graph $B = (V, P, Q)$ by Definition 2.1. We observe that it is easy to construct a suitable partition using a sequence that consists of 'double-leaf attachment' and 'edge-split' operation. Here, we only have to maintain two disjoint spanning trees according to the proof of Theorem 2.6. Hence, we obtain the following corollary.

Corollary 2.7. *Given a bispannable graph $G = (V, E)$ and a construction sequence consisting of 'double-leaf attachment' and 'edge-split' operations, it is possible to compute a partition of E into two spanning trees in time $\mathcal{O}(n)$.*

If we do not know a construction sequence, we have to compute a partition from scratch. Two algorithms for finding two edge-disjoint spanning trees in general graphs were given by Kameda and Toida [KT73]. Their runtime is $\mathcal{O}(\max\{n^2 \log n, mn\})$ and $\mathcal{O}(mn \log^* n)$, respectively. Here, it holds that $\log^* n = \min\{i \mid i \in \mathbb{N} \text{ and } \log^i n \leq 1\}$. Some properties which can be used to compute so-called maximally distant trees (two trees such that the number of common edges is minimal) were considered by Kishi and Kajitani [KK67; KK68; KK69]. Using an algorithm for computing k edge-disjoint spanning trees by Imai [Ima83], it is possible to find a partition of a bispannable graph in time $\mathcal{O}(n^2 \log n)$. Moreover, there exists an algorithm by Roskind and Tarjan [RT85] using time $\mathcal{O}(n^2)$. An improved version of this algorithm was presented by Gabow and Stallmann [GS85] as well as Gabow and Westermann [GW92]. These algorithms can compute a partition of a bispannable graph into two spanning trees in time $\mathcal{O}(n\sqrt{n \log n})$. We remark that the latter algorithms in [Ima83; RT85; GS85; GW92] are designed to compute a set of k edge-disjoint spanning trees or forests, respectively. For example, the algorithm by Roskind and Tarjan [Ros83; RT85] computes k edge-disjoint spanning trees in time $\mathcal{O}(k^2 n^2)$. A complexity survey of algorithms for solving so-called tree packing and covering problems can be found in [Sch03].

Most algorithms for computing k edge-disjoint spanning trees use so-called *augmenting swap sequences*. These sequences were introduced by Edmonds in the context of matroids [Edm65a; Edm65b] (see also [Law76]). A variant of them was also used by Hopcroft and Karp for computing a maximum matching in bipartite graphs [HK73]. Given an arbitrary bispannable graph $G = (V, E)$, the idea is to maintain two edge-disjoint forests, F_0 and F_1, in each round of the computation. A round consists of two steps: First, we try to compute an augmenting swap sequence for an edge $e \in E$. Subsequently, we perform an augmentation if such a sequence exist. Otherwise, if there is no such sequence, it is possible to store information about it for later computations (Roskind and Tarjan introduced the term 'clump', which is a set of vertices already connected in each of the forests [RT85]). The correctness of this approach follows because of the observation that the edges of G form a matroid (cf. Chapter 3) if we define a subset of edges $E' \subseteq E$ to be independent if E' can be partitioned into k forests [CH80]. Hence, we can apply the greedy algorithm.

More precisely, for the problem of computing a partition of a bispannable graph into two spanning trees, we initially start with $F_0 = F_1 = \emptyset$. Then, we compute an augmenting swap sequence for each edge $e \in E$ if such a sequence exists. Here, we remark that there always exists such a sequence if we consider a bispannable graph. An *augmenting swap sequence* with respect to an edge e and two forests F_0 and F_1 is formally defined to be a sequence of edges $e_0, e_1, \ldots, e_k$ such that

(i) $e_0 = e$,

(ii) $e_{i+1} \in C(F_{i \bmod 2}, e_i)$ for all $i = 0, \ldots, k-1$, and

(iii) there are no edges e_i and e_j with $j > i+1$ and $e_j \in C(F_{i \bmod 2}, e_i)$

where $C(F, e)$ denotes the cycle defined by e with respect to F (provided that the vertices of e are in the same tree with respect to F). Condition (iii) requires that the sequence has to be minimal in terms of having no shortcut.

Hence, if the insertion of e_0 into F_0 creates a cycle, we transfer an edge e_1 of this cycle to F_1. If this operation yields a cycle in F_1, we remove an edge e_2 of this cycle and insert this edge into F_0. This procedure continues until an edge e_k is inserted into a forest and does not create a cycle. Checking whether an edge generates a cycle or not can easily be done by using a union-find data structure [Tar75]. Thus, if already $F_0 \cup \{e\}$ does not contain any cycle then e itself is an augmenting swap sequence.

There exist two similar strategies to find such an augmenting sequence. On the one hand, we have the so-called *breadth-first scanning*. This strategy is accomplished using a queue Q of labeled edges. The labeling is used to reconstruct the augmenting sequence after the scanning. First, we label e (with `null` since this edge is the end/beginning of the sequence) and insert it into Q. In each step, the algorithm removes the first edge e from Q and checks whether there exists a forest F_i such that $F_i \cup \{e\}$ does not contain a cycle (if $e \in F_j$ then we omit the index j). If there exists an index i such that $F_i \cup \{e\}$ does not contain a cycle, we have found an augmenting swap sequence and perform the augmentation. Otherwise, we label the (unlabeled) edges of each cycle defined by e and insert them into Q. This algorithm terminates if an augmenting sequence is found or if the queue Q becomes empty. In the latter case, there exists no augmenting swap sequence with respect to edge e [GW92].

On the other hand, we can apply *cyclic scanning*. This approach is similar to the breadth-first scanning. The difference is that if we process a labeled edge $e \in F_i$, we consider only the next forest F_{i+1} (modulo the number of forests). If $F_{i+1} \cup \{e\}$ does not contain a cycle, we have found an augmenting swap sequence. Otherwise, we label the (unlabeled) edges of the cycle $C(F_{i+1}, e)$ and insert them into the queue Q. Again, the algorithm terminates if an augmenting sequence is found or if the queue Q becomes empty.

We remark that the algorithm of Roskind and Tarjan [RT85] uses cyclic scanning. A variant of this algorithm to partition a bispannable graph is presented in more detail in Algorithm 1). The improvement of Gabow and Westermann [GW92] primarily consists of computing a maximal set of shortest augmenting swap sequences in combination with cyclic scanning. The approach of using a maximal set of augmenting sequences is similar to Dinitz' algorithm [Din70; Eve79] for the maximum flow problem.

Algorithm 1: Partitioning($G = (V, E)$)

```
Input: a bispannable Graph G = (V, E)
Output: a partition of G into two edge-disjoint spanning trees
begin
    initialize union-find data structures F_0, F_1
    initialize index[e] = −1 for all e ∈ E
    initialize label[e] = null for all e ∈ E
    foreach e = {v, w} ∈ E do
        p_0[·], p_1[·] := compute trees rooted at v with respect to F_0 and F_1
        initialize queue Q of edges
        Q.enqueue(e)
        while !Q.empty() do
            e' = {x, y} = Q.dequeue()
            i = (index[e'] + 1) mod 2
            if F_i.find(x) ≠ F_i.find(y) then
                F_i.union(x, y)
                while e' ≠ e do
                    index[e'] = (i + 1) mod k
                    e' = label[e']
                    i = index[e']
                index[e] = 0
            else
                if x == v or label[{x, p_i[x]}] ≠ null then
                    u = y
                else
                    u = x
                initialize stack S of edges
                while u ≠ v and label[{u, p_i[u]}] ≠ null do
                    S.push({u, p_i[u]})
                    u := p_i[u]
                while !S.empty() do
                    e'' = S.pop()
                    label[e''] = e'
                    Q.enqueue(e'')
    return (F_0, F_1)
end
```

The Shannon Switching Game

The computation of two edge-disjoint spanning trees is strongly related to the so-called *Shannon switching game* [Sha55]. This game is sometimes also called *Lehman's switching game* because Lehman was the first one who found a solution [Leh64]. This game is played on an undirected connected graph in which two vertices are distinguished. There are two players which we call *Connector* and *Cutter*. In the literature, they are often called the *short* and *cut* player, respectively. The Cutter tries to destroy all paths between both distinguished vertices by removing edges whereas the Connector aims to establish a path of invulnerable edges between these two vertices. The first player who reaches his goal will win the game. The game is played alternately. In his turn, the Cutter chooses an unmarked edge which is removed from the graph. The Connector, in his turn, can make an unplayed edge invulnerable to deletion by the Cutter. A 'repairing' of previously deleted edges is not allowed.

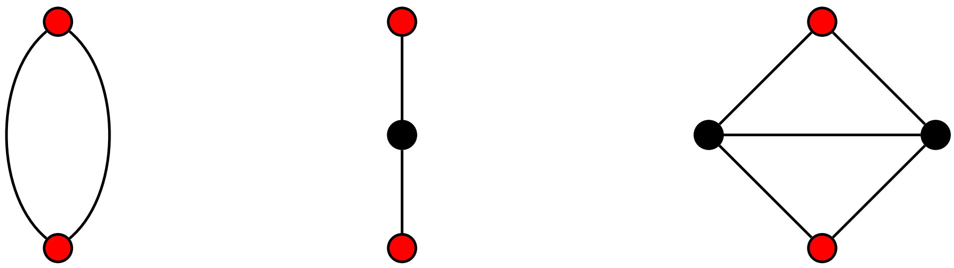
Depending on the player and who plays first, we obtain 3 different types of games.

(i) If the Connector, playing second, can win against all possible strategies of the Cutter, the game is called *short*.

(ii) If the Cutter, playing second, can win against all possible strategies of the Connector, the game is called *cut*.

(iii) If the player who plays first, but not second, can win against all possible strategies of the other player, the game is called *neutral*.

Note that if a player has a winning strategy when playing second, the player has a winning strategy when going first, too. This is clear since it is no handicap to have an extra move. Depending on which two vertices v and w are distinguished, a graph can yield a short, a cut, or a neutral game. Graphs for each of the three types are illustrated in Figure 2.12 where both vertices of the game are colored red.



(a) short (b) cut (c) neutral

Figure 2.12: Different graphs for the Shannon switching game.

As aforementioned, Lehman [Leh64] was the first who found a solution for the Shannon switching game. He analyzed the game using a matroid-theoretic approach like Edmonds who further studied it [Edm65a]. As a main result, Lehman showed that a graph yields a short game if and only if there exists a subgraph consisting of two edge-disjoint spanning trees such that both distinguished vertices are contained in this subgraph. Hence, if a graph $G = (V, E)$ contains two edge-disjoint spanning trees then for any two vertices $v, w \in V$, the corresponding game is a short game.

For the sake of completeness, a game is a cut game if and only if the corresponding game on the graph $G = (V, E \cup \{v, w\})$ is not a short game. Furthermore, a game is a neutral game if and only if it is not a short game and the corresponding game on $G = (V, E \cup \{v, w\})$ is a short game. For example, the graph in Figure 2.12(c) is neutral because the first player (Cutter or Connector) has to choose the edge connecting the black vertices in order to win the game. If we insert an edge joining the red vertices, there exists a winning strategy for the Connector.

A second approach to analyze the game was followed by Kishi and Kajitani [KK69]. They showed that the edges of a graph G can be decomposed into a partition containing three blocks which they call the *principal partition* of G. Bruno and Weinberg generalized the principal partition of a graph and showed that the type of a game can be determined immediately from this partition [BW70; BW71b]. The Shannon switching game and the principal partition of a graph were further analyzed in several articles and books [Bru74; Bru77; Cha72; Man96; Wei97; NG99].

Chapter 3

Ranking of Weighted Spanning Trees

3.1 Kano's Conjectures

In this chapter, we present several conjectures attributed to Kano [Kan87] as well as Mayr and Plaxton [MP92] concerning distances between spanning trees of different weights in the tree graph.

Recall that in the tree graph of a graph $G = (V, E)$, each spanning tree of G is represented by a vertex and two of these 'spanning tree' vertices are connected by an edge if and only if they are related by an edge swap. Given a weight function $w\colon E \to \mathbb{R}$, which associates a weight $w(e)$ with each edge $e \in E$, it is possible to get different spanning tree weights. Thus, we can partition the spanning trees of a graph into distinct weight classes.

Given a weighted undirected connected graph $G = (V, E, w)$ and its set of all spanning trees $\mathcal{T}(G)$, we denote by $\mathcal{W}(G)$ the set of different weights of spanning trees of G and by $\mathcal{W}_i(G)$ the ith smallest element of $\mathcal{W}(G)$. Analogously, we denote by $\mathcal{T}_i(G)$ the set of spanning trees T having $w(T) = \mathcal{W}_i(G)$. We define the order $\mathrm{ord}(G, T)$ of a spanning tree T with respect to G as the number $i \in \mathbb{N}$ such that $T \in \mathcal{T}_i(G)$. We denote the number of spanning trees with weight $w(T)$ by $\sigma(G, T)$, i.e., $\sigma(G, T) = |\mathcal{T}_{\mathrm{ord}(G,T)}(G)|$. Furthermore, we denote by $L_k(G, T)$ the set of all those spanning trees T' of G such that T can be transformed into T' by at most k edge swaps, i.e., the symmetric difference $T \Delta T' = (T \setminus T') \cup (T' \setminus T)$ contains at most $2k$ edges.

The following four conjectures were proposed by Kano [Kan87] motivated by a paper by Kawamoto, Kajitani, and Shinoda [KKS78]. Note that several authors (e.g., Kano in [Kan87]) formulated their results in terms of kth *maximal* spanning trees which is completely equivalent. We remark that the first of the following four conjectures actually is a theorem since Mayr and Plaxton found a proof [MP92].

Conjecture 1. *If T is a minimum spanning tree of G then $L_{i-1}(G, T)$ contains an ith smallest spanning tree for all $1 \leq i \leq |\mathcal{W}(G)|$.*

Conjecture 2. *If T is an ith smallest spanning tree in the set $L_i(G, T)$ then T is an ith smallest spanning tree of G.*

Conjecture 3. *If T is an ith smallest spanning tree of G then T is an ith smallest spanning tree in the set $L_{i-1}(G, T)$.*

Conjecture 4. *Let $G(i, j)$ denote the graph with vertex set $\mathcal{T}_i(G)$ where an edge exists between each pair of i-MSTs T and T' if T can be transformed into T' by at most j edge swaps in G. Then $G(i, i)$ is connected.*

It is well known that Conjectures 2 and 4 hold for $i = 1$ [Kru56; Pri57; FF62; Deo74]. Clearly, it holds that a spanning tree T is a 1-MST of a graph G if and only if it is a minimum spanning tree in $L_1(G, T)$. Moreover, let T and T' be two distinct minimum spanning trees. We claim that there exists a sequence $T = T_1, T_2, \ldots, T_k = T'$ consisting only of 1-MSTs such that T_i and T_{i+1} are related by an edge swap for all $i = 1, \ldots, k-1$. If $k = 2$ then we are done. Hence, we suppose that $k > 2$ and consider any edge $e \in T \setminus T'$. Insertion of e into T' generates a cycle $C(T', e)$. This cycle may contain edges in $T \cap T'$ but there is at least one edge $f \in F$ with $F = C(T', e) \setminus T$. Now, it holds that $w(e) \geq w(f)$ for all $f \in F$ since otherwise T' is not a minimum spanning tree. We claim that at least one of these edges $f \in F$ has the same weight as the edge e. This follows from the observation that at least one edge in F has to cross the cut formed by e with respect to T. If this edge has a weight strictly smaller than $w(e)$ then $w(T \setminus \{e\} \cup \{f\}) < w(T)$ in contradiction to T being a 1-MST. Hence, there exists an edge $f \in F$ with weight $w(f) = w(e)$. Now, the spanning tree $T'' = (T' \cup \{e\}) \setminus \{f\}$ is a minimum spanning tree, too. Moreover, T'' has one edge more in common with T (than T'). We repeat this procedure and obtain a series of spanning trees $T = T_1, T_2, \ldots, T_k = T'$ where two consecutive trees are related by an edge swap.

The above statements follow easily by using a special edge swap regarding two distinct spanning trees. Let $G = (V, E)$ be an undirected graph, T and T' be two distinct spanning trees, and $e \in T \setminus T'$. Then, there exists an edge $f \in T' \setminus T$ such that $(T \cup \{f\}) \setminus \{e\}$ *and* $(T' \cup \{e\}) \setminus \{f\}$ are spanning trees. In this case, the edge swap (e, f) (and its inverse (f, e)) with respect to T and T' is called a *symmetric exchange* [Bru69]. The existence of such an exchange can be proven as follows. If we choose an arbitrary edge $e \in T$ then $T \setminus \{e\}$ (the corresponding subgraph) decomposes into two connected components C_1 and C_2. Color the vertices of these two components with different colors. Note that the two vertices of edge e have different colors. Now we consider the fundamental cycle $C(T', e)$ defined by e with respect to T' and choose an edge $f \in T'$ such that the two vertices of f have different colors. Clearly, such an edge must exist, and $(T \cup \{f\}) \setminus \{e\}$ as well as $(T' \cup \{e\}) \setminus \{f\}$ are spanning trees. Symmetric exchanges were further studied by Brylawski [Bry73] and White [Whi80]. Note that there exists a generalized version of symmetric exchanges for whole subset. This version is known as the *symmetric subset exchange axiom* and was formulated in matroid theory.

Lemma 3.1. *Let B, B' be bases of a matroid. If B is partitioned into X and Y, then there is a partition of B' into X' and Y' such that $X \cup Y'$ and $Y \cup X'$ are bases, too.*

For a proof of this lemma, we refer the reader to [Sch03]. Several derivations of it with respect to matroid theory are considered in [Bry73; Gre73; Woo74]. A stronger exchange property called 'strongly base orderable' is studied in Section 3.3.2.

Kawamoto, Kajitani, and Shinoda [KKS78] proved that Conjectures 1 through 4 hold if $i = 2$. Kano simplified the results for $i = 2$ and extended them for proving the case $i = 3$. He made use from a special property: for each minimum spanning tree T and an arbitrary spanning tree T', there exists a bijection $\varphi\colon T \setminus T' \to T' \setminus T$ such that $T'' = (T' \setminus \varphi(e)) \cup \{e\}$ is a spanning tree and $w(T'') \leq w(T')$ for each edge $e \in T \setminus T'$.

As already mentioned above, Conjecture 1 is actually a theorem. A powerful tool for proving this conjecture are contractions and deletions of edges. Let $G = (V, E)$ be a graph and $e, f \in E$ be two distinct edges. We denote by $G[e, f]$ the graph we obtain by contracting the edge e and deleting the edge f. According to our notation of Chapter 2, it holds $G[e, f] = (G \setminus f)/e$. Some properties which will be used throughout this chapter are summarized in the following lemma [MP92].

Lemma 3.2. *Let T be a spanning tree of a weighted graph $G = (V, E, w)$, and let $e \in E$. If $e \notin T$, let $G' = G[\emptyset, e]$ and $T' = T$. Otherwise, let $G' = G[e, \emptyset]$ and $T' = T[e, \emptyset]$. In either case, it holds that:*

(i) T' *is a spanning tree of* G'.

(ii) $\operatorname{ord}(G', T') \leq \operatorname{ord}(G, T)$.

(iii) $\sigma(G', T') \leq \sigma(G, T)$.

Proof. The first condition is obvious: If e belongs to T, the contraction of e preserves the property of being a spanning tree. Furthermore, removing an edge not contained in T has no impact, too. For the remaining two cases, we observe that there exists an injective map from $\mathcal{T}(G')$ to $\mathcal{T}(G)$ satisfying a constant shift of the weight classes depending on $w(e)$. If $e \notin T$ then each spanning tree in $\mathcal{T}(G')$ is also a spanning tree in $\mathcal{T}(G)$ (especially, each spanning tree with a weight smaller than $w(T)$). Hence, condition (ii) and (iii) follow if $e \notin T$. Now, we suppose that $e \in T$. In this case, each spanning tree of $\mathcal{T}(G')$ is mapped to a spanning tree in $\mathcal{T}(G)$ with a weight increased by $w(e)$. This proves the lemma. □

In their proof of Conjecture 1, Mayr and Plaxton used bispanning graphs. More precisely, they showed that this conjecture is true if and only if there is no weighted bispanning graph $B = (V, P, Q, w)$ such that $1 = \operatorname{ord}(B, P) < \operatorname{ord}(B, Q) < n$ and $\sigma(B, Q) = 1$. Indeed there is no such bispanning graph.

Theorem 3.3. *There exists no weighted bispanning graph $B = (V, P, Q, w)$ such that $1 = \operatorname{ord}(B, P) < \operatorname{ord}(B, Q) < n$ and $\sigma(B, Q) = 1$.*

Proof. Suppose that there exists such a weighted bispanning graph. Then, consider any smallest (with respect to $|V|$) of these graphs $B = (V, P, Q, w)$ with $1 = \text{ord}(B, P) < \text{ord}(B, Q) < n$ and $\sigma(B, Q) = 1$. Let p be an edge with maximum weight in P and let $q \in Q$ be an edge such that (p, q) is a symmetric exchange with respect to P and Q. As mentioned before (and by Lemma 3.1), there always exists such an edge q. Since P is a minimum spanning tree and Q has unique weight, we have $w(p) < w(q)$. Moreover, since p has maximum weight in P, the edge q is the unique heaviest in the cycle $C(P, q)$. Hence, q does not belong to any minimum spanning tree by the *red rule* [Tar83].

Now, we consider the bispanning graph $B' = (V', P', Q', w') := B[q, p]$ with $P' = P[q, p]$ and $Q' = Q[q, p]$. By Lemma 3.2, we have $\text{ord}(B', P') = 1$ and $\sigma(B', Q') = 1$, thus, $\text{ord}(B', P') < \text{ord}(B', Q')$. Since the number of vertices is decreased by one, we have to show that $\text{ord}(B', Q') < \text{ord}(B, Q)$ (by Lemma 3.2, we only have '$\leq$'). In this case, we have a strict inequality because no spanning tree of B' is mapped into the minimum spanning tree class $\mathcal{T}_1(B)$. ☐

Mayr and Plaxton not only proved Conjecture 1 but also formulated a further conjecture which unifies Kano's Conjectures 2 through 4. This new conjecture is as follows.

Conjecture 5. *If T is a jth smallest spanning tree of G then $L_{i-1}(G, T)$ contains an ith smallest spanning tree for all $1 \leq j < i \leq |\mathcal{W}(G)|$.*

Analogously to the ideas above, a stronger version of Theorem 3.3 would imply that Conjecture 5 is a theorem. This stronger version is as follows: there exists no weighted bispanning graph $B = (V, P, Q, w)$ such that $\text{ord}(B, P) < \text{ord}(B, Q) < n$ and $\sigma(B, Q) = 1$. Hence, we would be done by proving the following conjecture which implies Conjectures 2 through 5.

Conjecture 6. *Let $B = (V, P, Q, w)$ be a weighted bispanning graph such that $\text{ord}(B, P) < \text{ord}(B, Q)$ and $\sigma(B, Q) = 1$. Then, it holds that $\text{ord}(B, Q) \geq n$.*

In the subsequent section, we show that this is true if P has unique weight, too, i.e., we have $\sigma(B, P) = 1$. If P's weight is not unique, it might be sufficient to count so-called partition spanning trees corresponding to subsequence-interchangeable base orderings (cf. Chapters 4 and 5).

3.2 Assuming Singularity of P

In this section, we prove Conjecture 6 under the assumption that the spanning tree P is unique, i.e., the weight function is also restricted to satisfy $\sigma(B, P) = 1$.

Theorem 3.4. *Let $B = (V, P, Q, w)$ be a weighted bispanning graph such that $\text{ord}(B, P) < \text{ord}(B, Q)$, $\sigma(B, Q) = 1$, and $\sigma(B, P) = 1$. Then, it holds that $\text{ord}(B, Q) \geq n$.*

Proof. This Theorem is proven by induction over the number of vertices of B. Clearly, if $n = 2$ then B consists of two parallel edges with distinct weights, thus $\text{ord}(B, Q) \geq n = 2$. Hence, we assume $n > 2$ and consider an arbitrary symmetric exchange (p, q) with $p \in P$ and $q \in Q$, that is, $P \setminus \{p\} \cup \{q\}$ and $Q \setminus \{q\} \cup \{p\}$ are spanning trees. By Lemma 3.1, such a symmetric exchange exists. Since $\sigma(B, Q) = 1$ (or $\sigma(B, P) = 1$), the edges p and q must have different weight. Now, we distinguish different cases depending on whether $w(p) < w(q)$ or $w(q) > w(p)$ holds.

1. If $w(q) < w(p)$ holds then we consider the bispanning graph $B' = B[q, p]$ which we obtain by contracting the edge q and removing the edge p in B. The elements of $\mathcal{W}(B')$ in strictly increasing order are (for the sake of readability, we associate by a spanning tree T also its weight $w(T)$)

$$(T'_1, \ldots, T'_{\alpha-1}, T'_\alpha = P', T'_{\alpha+1}, \ldots, B'_{\beta-1}, T'_\beta = Q', T'_{\beta+1}, \ldots, T'_\gamma)$$

   with weights $T'_\alpha = P' = P[q, p]$ and $T'_\beta = Q' = Q[q, p]$. By Lemma 3.2, it holds that $\sigma(B', Q') = 1$, thus, applying the induction hypothesis, we obtain $\text{ord}(B', Q') = \beta \geq n - 1$. We observe that each spanning tree of B' together with the edge q forms a spanning tree of B. Thus, there are at least β spanning trees having the distinct weights

$$(T_1, \ldots, T_{\alpha-1}, T_\alpha, T_{\alpha+1}, \ldots, T_\beta)$$

   with weights $T_i = T'_i + w(q)$, $1 \leq i \leq \beta$, such that each of these weights is smaller than or equal to $w(Q)$. Since $\sigma(B, P) = 1$, none of these spanning tree weights can map into the weight $w(P)$. Hence, we obtain $\text{ord}(B, Q) \geq \beta + 1 \geq n$ since $w(P) < w(Q)$. This situation is illustrated in Figure 3.1 where P' is mapped to the lower spanning tree P since $w(q) < w(p)$.

2. We assume $w(p) < w(q)$ and

   (a) $w(P) - w(p) < w(Q) - w(q)$. Analogous to the first case, we consider the bispanning graph $B' = B[q, p]$ and the elements of $\mathcal{W}(B')$ in strictly increasing order

$$(T'_1, \ldots, T'_{\alpha-1}, T'_\alpha = P', T'_{\alpha+1}, \ldots, T'_{\beta-1}, T'_\beta = Q', T'_{\beta+1}, \ldots, T'_\gamma) \, .$$

   Once more, we have $\sigma(B', Q') = 1$ by Lemma 3.2. Applying the induction hypothesis, we obtain $\text{ord}(B', Q') = \beta \geq n - 1$. Again, each spanning tree of B' can be combined with the contracted edge q obtaining the following weights

$$(T_1, \ldots, T_{\alpha-1}, T_\alpha, T_{\alpha+1}, \ldots, T_\beta)$$

   with $T_i = T'_i + w(q)$, $1 \leq i \leq \beta$, (see Figure 3.1 with the upper spanning tree P). Each of these distinct weights is smaller than or equal to $w(Q)$ and none of them can map into the weight $w(P)$. Hence, we also count the weight of spanning tree P resulting in $\text{ord}(B, Q) \geq \beta + 1 \geq n$.

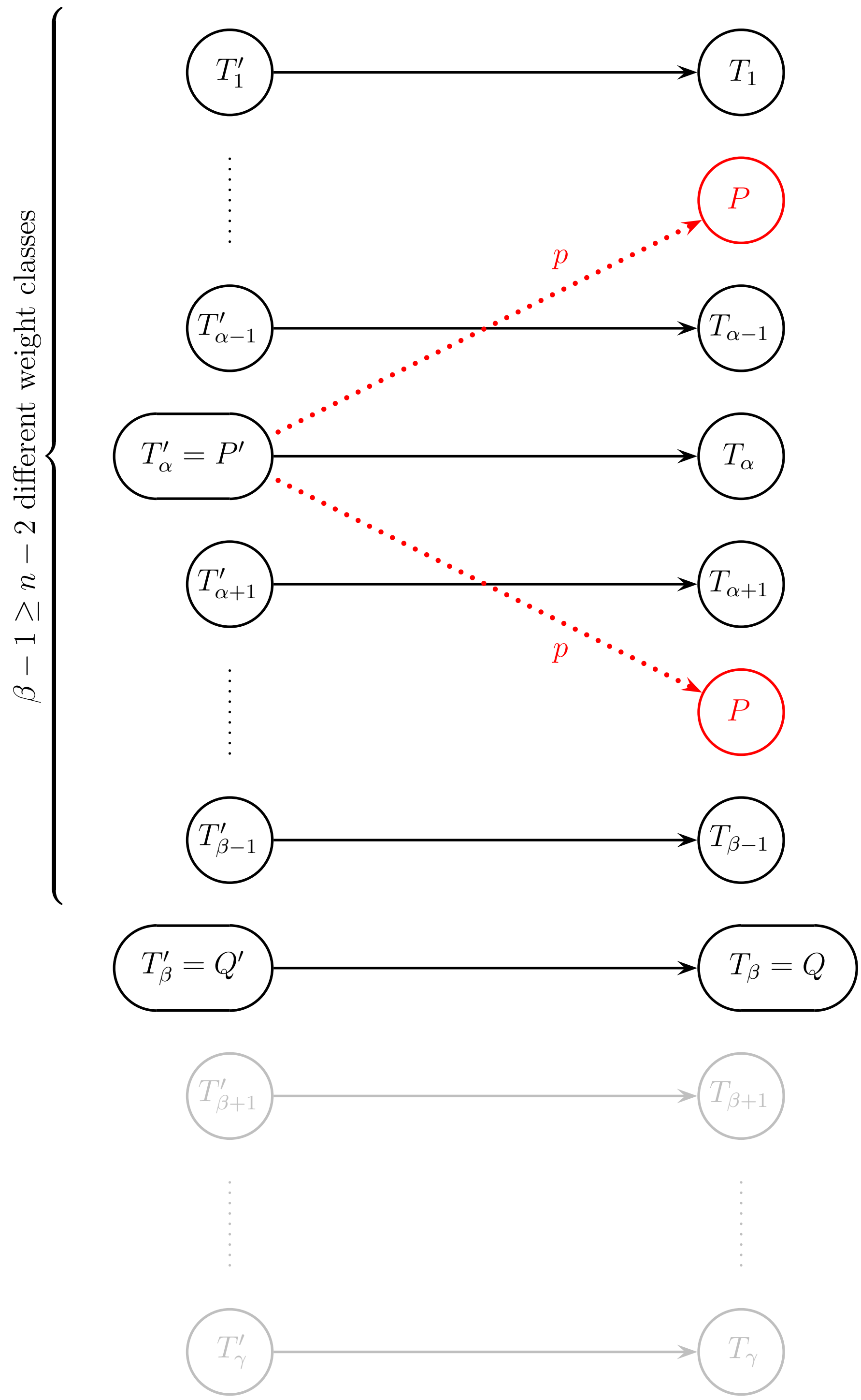


Figure 3.1: Constructing new classes of spanning trees by adding the edge q (solid lines) to the classes of $B[q,p]$. Depending on whether $w(q) < w(p)$ or $w(p) < w(q)$ (together with $w(P') < w(Q')$) holds we get either the upper or the lower spanning tree P.

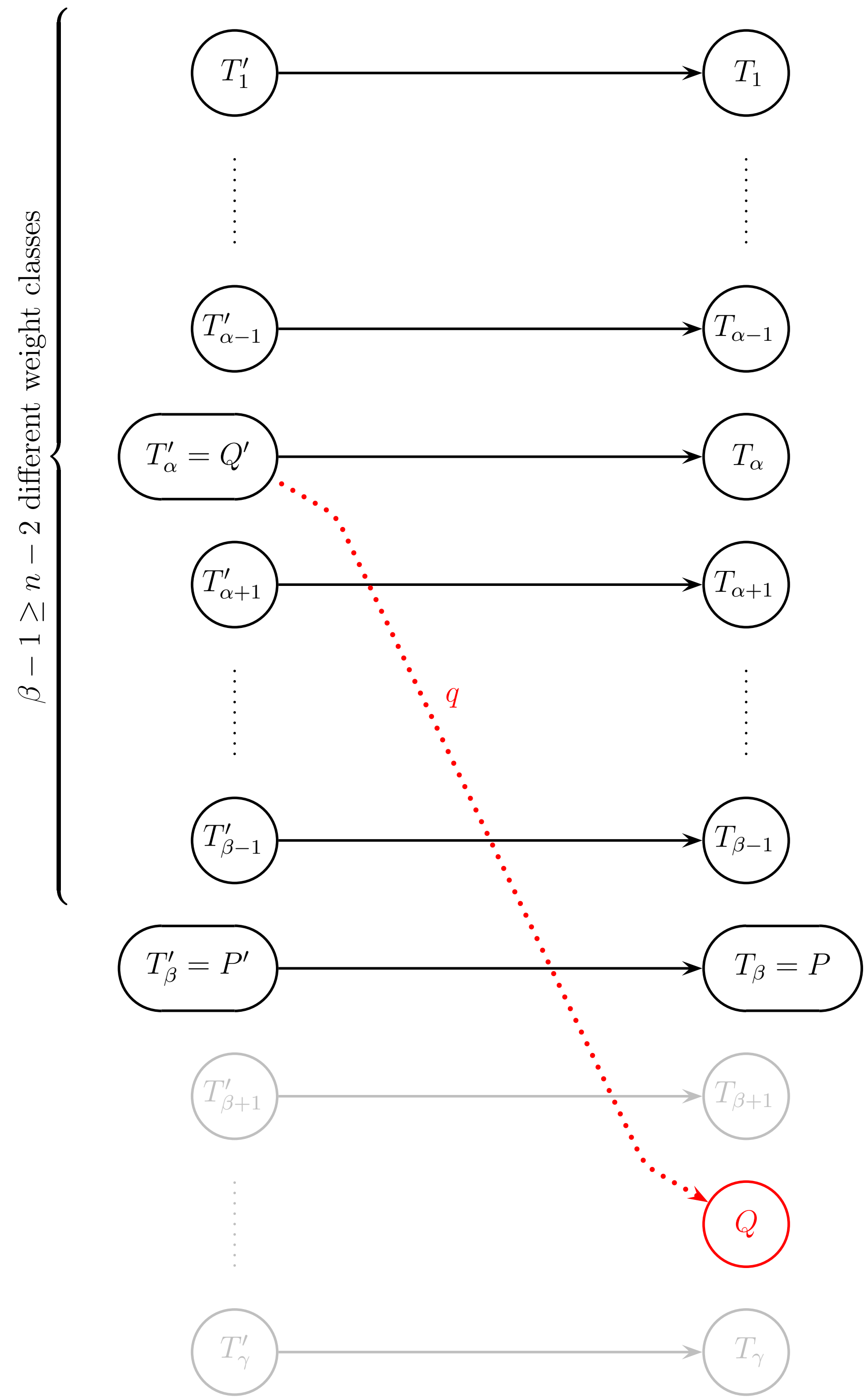


Figure 3.2: Constructing new classes of spanning trees by adding the edge p (solid lines) to the classes of $B[p, q]$.

(b) $w(Q)-w(q) < w(P)-w(p)$. The main difference to the previous two cases is that we exchange the role of P and Q by contracting p and removing q, that is, we consider the bispanning graph $B' = B[p,q]$. The increasing sequence of weights of $\mathcal{W}(B')$ is

$$(T'_1, \dots, T'_{\alpha-1}, T'_\alpha = Q', T'_{\alpha+1}, \dots, T'_{\beta-1}, T'_\beta = P', T'_{\beta+1}, \dots, T'_\gamma) \ .$$

By Lemma 3.2, it holds that $\sigma(B', P') = 1$. Thus, we apply the induction hypothesis and obtain $\operatorname{ord}(B', P') = \beta \geq n-1$. Combining these spanning trees in B' with the contracted edge p, we get at least $\beta \geq n-1$ different spanning trees with distinct weights

$$(T_1, \dots, T_{\alpha-1}, T_\alpha, T_{\alpha+1}, \dots, T_\beta, P)$$

where $T_i = T'_i + w(p)$, $1 \leq i \leq \beta$. This situation is illustrated in Figure 3.2. Since $w(P) < w(Q)$, we obtain $\operatorname{ord}(B, Q) \geq \beta + 1 \geq n$ and the theorem follows. □

3.3 An Analysis using Matroid Theory

In this section, we give a short introduction to matroid theory in order to use it for the analysis of bispanning graphs. In the first part, we introduce basic concepts and several different descriptions of matroids. For a deeper view into matroid theory, we refer the reader to the most common literature about matroids [Tut71; Wel76; Oxl92; Sch03]. After this introduction, we use these concepts to lower bound the number of spanning trees with distinct weights of a bispanning graph. In this analysis, we restrict our analysis to bispanning graphs where the set of all acyclic subgraphs is a special (a so-called strongly base orderable) matroid. Note that Mayr and Plaxton already considered this graph class (they used the notation of a 'parallel swap' [MP92]) but they did not establish the connection to matroid theory. Furthermore, we show that each bispanning graph can be analyzed in this way if it contains no minor isomorphic to the K_4.

3.3.1 Preliminaries

As we will see, there exist different definitions of a matroid. Whitney, who firstly used the term 'matroid' in his seminal paper on the abstract properties of linear dependence [Whi35], already gave several descriptions. One of the most common definitions is the following:

Definition 3.5. *A pair $(E, \mathcal{I})$ consisting of a finite set E and a non-empty family $\mathcal{I}$ of subsets of E is called a* matroid *if $\mathcal{I}$ satisfies the following three conditions:*

(I1) $\emptyset \in \mathcal{I}$,

(I2) $I_1 \in \mathcal{I}$ *and* $I_2 \subseteq I_1$ *imply* $I_2 \in \mathcal{I}$, *and*

(I3) for all $I_1, I_2 \in \mathcal{I}$ *with* $|I_1| < |I_2|$, *there exists some* $e \in I_2 \backslash I_1$ *such that* $I_1 \cup \{e\} \in \mathcal{I}$.

Let $M = (E, \mathcal{I})$ be a matroid. A subset I of E is called *independent* if $I \in \mathcal{I}$, and *dependent* otherwise. An independent subset B of E is called a *base* if there is no independent subset B' of E such that $B \subset B'$. As a consequence of condition *(I3)*, all bases of a matroid $M = (E, \mathcal{I})$ have same cardinality. We denote the collection of bases of M by $\mathcal{B}(M)$. If the matroid M is clear from the context, we will shortly write $\mathcal{B}$. Contrary to bases, a dependent set of minimum size is called a *cycle*. The collection of cycles of a matroid M is denoted by $\mathcal{C}(M)$ or short by $\mathcal{C}$.

Another way to define a matroid M is by its collection of bases $\mathcal{B}(M)$.

Proposition 3.6. *Let $\mathcal{B}$ be a set of subsets of a set E and let $\mathcal{I}$ be the set of all subsets of E that are contained in some $B \in \mathcal{B}$. Then $\mathcal{B}$ is the collection of bases of a matroid $M = (E, \mathcal{I})$, if and only if the following conditions are satisfied:*

(B1) $\mathcal{B} \neq \emptyset$ *and*

(B2) *for any two $B_1, B_2 \in \mathcal{B}$ and $e \in B_1 \setminus B_2$, there exists an element $f \in B_2 \setminus B_1$ such that $(B_1 \setminus \{e\}) \cup \{f\} \in \mathcal{B}$.*

Proof. We start by proving the 'only if' direction. Because E is a finite set, the set $\mathcal{I}$ is finite and contains at least one element by condition *(I1)*. Now, let $B_1, B_2 \in \mathcal{B}$ and $e \in B_1 \setminus B_2$. We consider $I_1 := B_1 \setminus \{e\}$ and $I_2 := B_2$. Since all bases have same cardinality, it holds that $|I_2| = |I_1| + 1$, thus, by *(I3)*, there exists $f \in I_2 \setminus I_1$ such that $B' = (B_1 \setminus \{e\}) \cup \{f\} \in \mathcal{I}$ implying $B' \in \mathcal{B}$.

To establish the 'if' direction, let $\mathcal{B}$ be a set of subsets of E satisfying *(B1)* and *(B2)*. Let $\mathcal{I}$ be the set of subsets of E that are contained in some $B \in \mathcal{B}$. We have to show that $M = (E, \mathcal{I})$ is a matroid. Since $\mathcal{B}$ is non-empty, $\mathcal{I}$ satisfies *(I1)*. Let $I_1 \in \mathcal{I}$ implying $I_1 \subseteq B$ for some set $B \in \mathcal{B}$. Hence, $I_2 \subseteq I_1$ implies $I_2 \subseteq \mathcal{B}$ and so $I_2 \in \mathcal{I}$. For proving *(I3)*, we claim that all sets B of $\mathcal{B}$ have the same cardinality. To this end, we suppose there exist sets B_1 and B_2 with $|B_1| > |B_2|$ and $|B_1 \setminus B_2| > 0$ (minimal with respect to the latter property). By *(B2)*, for an element $e \in B_1 \setminus B_2$ there exists some $f \in B_2 \setminus B_1$ such that $(B_1 \setminus \{e\}) \cup \{f\} \in \mathcal{B}$ contradicting the minimality of $|B_2 \setminus B_1| > 0$. We assume there are two sets $I_1, I_2 \in \mathcal{I}$ with $|I_1| < |I_2|$ such that for all $e \in I_2 \setminus I_1$ the set $I_1 \cup \{e\}$ does not belong to $\mathcal{I}$. By construction of $\mathcal{I}$, there are sets B_1 and B_2 of $\mathcal{B}$ such that $I_1 \subseteq B_1$ and $I_2 \subseteq B_2$. We choose B_2 minimizing $|B_2 \setminus (I_2 \cup B_1)|$. Since *(I3)* fails for I_1 and I_2, it holds that

$$I_2 \setminus B_1 = I_2 \setminus I_1 \ . \tag{3.1}$$

We claim that $B_2 \setminus (I_2 \cup B_1)$ is empty. Otherwise let $e \in B_2 \setminus (I_2 \cup B_1)$. Then, by *(B2)*, there exists an element $f \in B_1 \setminus B_2$ such that $(B_2 \setminus \{e\}) \cup \{f\} \in \mathcal{B}$ contradicting the choice of B_2. Therefore, we obtain $B_2 \setminus B_1 = I_2 \setminus B_1$ and

$$B_2 \setminus B_1 = I_2 \setminus I_1 \ . \tag{3.2}$$

Now, we claim that $B_1 \setminus (I_1 \cup B_2)$ is empty. Otherwise, for an arbitrary element e in this set, there exists an element $f \in B_2 \setminus B_1$ such that $(B_1 \setminus \{e\}) \cup \{f\} \in \mathcal{B}$. Since

$I_1 \cup \{f\} \subseteq (B_1 \setminus \{e\}) \cup \{f\}$, the set $I_1 \cup \{f\}$ is independent. Furthermore (3.2) implies $f \in I_2 \setminus I_1$ contradicting our assumption that I_1 and I_2 is a counterexample for *(I3)*. Therefore, we obtain $B_1 \setminus B_2 = I_1 \setminus B_2$ implying

$$B_1 \setminus B_2 \subseteq I_1 \setminus I_2 \, . \tag{3.3}$$

Because of the equicardinality of all $B \in \mathcal{B}$, (3.2), (3.3), we obtain $|I_1 \setminus I_2| \geq |I_2 \setminus I_1|$ implying $|I_1| \geq |I_2|$ which is a contradiction to our assumption. Hence, $M = (E, \mathcal{I})$ is a matroid. □

A matroid M can also be characterized by its collection of cycles (without a proof).

Proposition 3.7. *Let $\mathcal{C}$ be a set of subsets of a set E and let $\mathcal{I}$ be the collection of subsets of E that contain no set of $\mathcal{C}$. Then $\mathcal{C}$ is the collection of cycles of a matroid $M = (E, \mathcal{I})$ if and only if the following conditions are satisfied:*

(C1) $\emptyset \notin \mathcal{C}$,

(C2) $C_1, C_2 \in \mathcal{C}$ *and* $C_1 \subseteq C_2$ *imply* $C_1 = C_2$,

(C3) $C_1, C_2 \in \mathcal{C}$ *and* $e \in C_1 \cap C_2$ *imply* $C_3 \subseteq (C_1 \cup C_2) \setminus \{e\}$ *for some* $C_3 \in \mathcal{C}$.

A consequence of this description by the collection of cycles is as follows. For each independent set $I \in \mathcal{I}$ of a matroid $M = (E, \mathcal{I})$ and an element $e \in E$ such that $I \cup \{e\}$ is a dependent set, the set $I \cup \{e\}$ contains a unique cycle, which is an element of $\mathcal{C}$. This cycle is called the *fundamental cycle* of I with respect to e.

A very famous matroid in graph theory is the so-called *cycle matroid* of a graph. In the literature [Tut65], this matroid is also often referred to as the *polygon matroid*.

Proposition 3.8. *Let $G = (V, E)$ be a graph and let $\mathcal{I}$ be the family of all subsets of E such that each $I \in \mathcal{I}$ is a acyclic subgraph (i.e. a forest). Then $M = (E, \mathcal{I})$ is a matroid which is called the* cycle matroid *of G.*

Proof. Clearly, conditions *(I1)* and *(I2)* hold. Let I_1 and I_2 be two acyclic subgraphs of G with $|I_1| < |I_2|$. For proving condition *(I3)*, we assume that I_1 and I_2 are forests with $|I_2| = |I_1| + 1$. This is no restriction since it is possible to remove arbitrary elements from I_2 obtaining an acyclic subgraph of G. Consider the induced graphs $G_1 = (V, I_1)$ and $G_2 = (V, I_2)$. The number of connected components in G_1 is $c_1 = |V| - |I_1|$ and $c_2 = |V| - |I_2|$ in G_2, respectively. Hence, it holds that $c_1 = c_2 + 1$. Thus, there are two vertices lying in different connected components with respect to G_1 which are connected by an edge in G_2. Now, choose an arbitrary edge e connecting these two components yielding the independent set $I_1 \cup \{e\}$. □

3.3.2 Transversal and Strongly Base Orderable Matroids

In this section, we introduce two special matroids. The first one which is called *transversal matroid* was firstly observed by Edmonds and Fulkerson [EF65]. Furthermore, there are several articles and books which study transversals (e.g., [BS68; Mir71; BD72]). Bondy and Welsh formulated an algorithm to test whether or not a given matroid is transversal [BW71a]. Afterwards, we consider matroids which have the property of being strongly base orderable.

Let E be a finite set and $\mathcal{A} = (A_1, \ldots, A_r)$ be an arbitrary collection of subsets of E. A subset $E' = \{e_1, \ldots, e_k\} \subseteq E$ of distinct elements is a *partial transversal* of $\mathcal{A}$ if there exists a subfamily $(A_{i_1}, \ldots, A_{i_k})$ of distinct sets of $\mathcal{A}$ such that $e_j \in A_{i_j}$ for all $1 \leq j \leq k$. The element e_j is said to *represent* the set A_{i_j} in E'. A partial transversal of $\mathcal{A}$ of size r is called a *transversal* of $\mathcal{A}$. Now, we are ready to define the class of transversal matroids.

Proposition 3.9. *Let $\mathcal{A}$ be a family of finite subsets of a set E and let $\mathcal{I}_{\mathcal{A}}$ be the collection of partial transversals of $\mathcal{A}$. Then, $M = (E, \mathcal{I}_{\mathcal{A}})$ is a matroid which is called transversal matroid.*

The collection $\mathcal{A}$ is said to be a *representation* of the matroid $M = (E, \mathcal{I}_{\mathcal{A}})$. We note that the representation of a transversal matroid is not necessarily unique.

In a natural way, each representation $\mathcal{A}$ of a transversal matroid $M = (E, \mathcal{I}_{\mathcal{A}})$ is related to a bipartite graph representation $G(E, \mathcal{A})$. The vertex set of this bipartite graph consists of vertices v_e for each $e \in E$ and vertices v_A for each set $A \in \mathcal{A}$. A vertex v_e representing the element $e \in E$ is connected to a vertex v_A if $e \in A$. For example, let $E = \{e_1, e_2, e_3, e_4, e_5\}$ and $\mathcal{A} = (A_1, A_2, A_3)$ with $A_1 = \{e_1, e_2, e_3\}$, $A_2 = \{e_2, e_3, e_5\}$, and $A_3 = \{e_4, e_5\}$, the associated bipartite graph representation is illustrated in Figure 3.3.

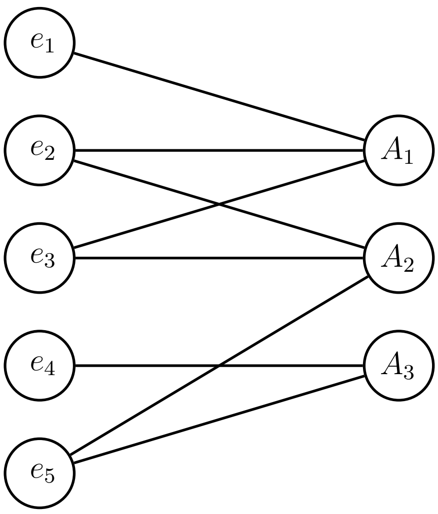


Figure 3.3: Bipartite graph representation of a transversal matroid.

Proof of Proposition 3.9 . Clearly, $\mathcal{I}_{\mathcal{A}}$ satisfies *(I1)* and *(I2)*. Now, let I_1 and I_2 with $|I_1| < |I_2|$ be partial transversals. The bipartite graph representation $G(E, \mathcal{A})$ contains two matchings M_1 and M_2 that match I_1 and I_2, respectively, with $\mathcal{A}$. Consider the graph induced by the the symmetric difference $M_1 \Delta M_2 = (M_1 \setminus M_2) \cup (M_2 \setminus M_1)$. This graph consists of cycles and paths where each cycle must have even length. Since the number of edges in $M_2 \setminus M_1$ is greater than the number of edges in $M_1 \setminus M_2$, there exists a path of odd length $P = (v_1, v_2, \ldots, v_{2k})$ which starts and ends with edges of M_2. Without loss of generality, we assume v_1 is a vertex representing an element $e \in E$. Then, $I_1 \cup \{v_1\}$ can be matched in $\mathcal{A}$. Hence, $\mathcal{I}$ satisfies *(I3)*. □

Now, we consider another specification of matroids. These matroids have the property to be strongly base orderable. Subsequently, we show that Conjecture 6 holds for bispanning graphs $B = (V, P, Q)$ whose cycle matroid is strongly base orderable.

Definition 3.10. *A matroid $M = (E, \mathcal{I})$ is called strongly base orderable if there exists a bijection $\varphi\colon B \to B'$ for each two bases B, B' such that for each subset X of B the set $(B \setminus X) \cup \varphi(X)$ is a base, too.*

A very useful property of strongly base orderable matroids is proven in the following lemma.

Lemma 3.11. *Let $M = (E, \mathcal{I})$ be a strongly base orderable matroid. Then for each two bases B and B' there exists a bijection $\varphi\colon B \to B'$ such that for all subsets X of B the sets $(B \setminus X) \cup \varphi(X)$ and $(B' \setminus \varphi(X)) \cup X$ are bases.*

Proof. By the definition of strongly base orderable matroids, there exists a bijection $\varphi\colon B \to B'$ such that for each $X \subseteq B$ the set $(B \setminus X) \cup \varphi(X)$ is a base. Let X be a subset of B of minimal cardinality such that $(B' \setminus \varphi(X)) \cup X$ is not a base, that is, $(B' \setminus \varphi(X)) \cup X$ contains exactly one cycle C since otherwise X is not minimal. Furthermore, because of this minimality we have $X \subset C$. Let $X' = C \setminus X$ and let $\tilde{X} = \varphi^{-1}(X') \subseteq B$ be the elements of B that map to an element of X'. Clearly, it holds that $X \cap \tilde{X} = \emptyset$. Furthermore, the set $(B \setminus \tilde{X}) \cup \varphi(\tilde{X})$ contains the cycle C in contradiction to the property of M to be strongly base orderable. □

Note that the bases B and B' do not have to be disjoint. In this case, the relation for all elements in $B \cap B'$ is the identity. However, in our analysis we only need the special case of disjoint bases denoted by P and Q.

Theorem 3.12. *Let $B = (V, P, Q)$ be a weighted bispanning graph such that $\operatorname{ord}(B, P) < \operatorname{ord}(B, Q)$ and $\sigma(B, Q) = 1$. Let $M = (P \cup Q, \mathcal{I})$ be the cycle matroid of B. If M is strongly base orderable then it holds that $\operatorname{ord}(B, Q) \geq n$.*

Proof. Since M is strongly base orderable, there exists a bijection $\varphi\colon Q \to P$ such that for each subset Q' of Q the set $(Q \setminus Q') \cup \varphi(Q')$ is a spanning tree. Let $w\colon (P \cup Q) \to \mathbb{R}$ be the weight function of B. Clearly, it holds that $w(Q') \neq w(\varphi(Q'))$ for each $Q' \subseteq Q$ since otherwise we have $\operatorname{ord}(B, Q) > 1$. Let $\delta(q) = w(\varphi(q)) - w(q)$ be the difference of the weights of an edge q and its image $\varphi(q)$ with respect to the weight function w. Thus, the function δ measures the increase of the spanning tree weight after replacing the edge q by its image $\varphi(q)$. In general, we define for a subset Q' of Q

$$\delta(Q') = \sum_{q \in Q'} \delta(q) = \sum_{q \in Q'} \big(w(\varphi(q)) - w(q)\big) \,. \tag{3.4}$$

If we choose $Q' = Q$, equation (3.4) implies

$$\delta(Q) = \sum_{q \in Q} \delta(q) = \sum_{q \in Q} \big(w(\varphi(q)) - w(q)\big) = w(P) - w(Q) < 0 \iff w(P) < w(Q) \,.$$

Now, we arrange the elements of Q in such a way that $\delta(q_1) \leq \ldots \leq \delta(q_{n-1})$ holds and consider the $n-1$ different sets $Q_i = \{q_1, \ldots, q_i\}$ for $1 \leq i \leq n-1$. Clearly, for each $1 \leq i \leq n-1$ the set $(Q \setminus Q_i) \cup \varphi(Q_i)$ is a spanning tree since M is a strongly base orderable matroid. Furthermore, each set Q_i satisfies

$$\delta(Q_i) = \sum_{q \in Q_i} \delta(q) < 0$$

that is, if we remove the edges Q_i from Q and add the edges $\varphi(Q_i)$, the weight of the resulting spanning tree is smaller than $w(Q)$. Thus, if we show that these spanning tree weights are distinct, the claim follows. Let i and j be two indices such that $\delta(Q_i) = \delta(Q_j)$. In this case, it holds that

$$\begin{aligned}
\delta(Q') &= \sum_{q \in Q'} \delta(q) = \sum_{q \in Q'} \big(w(\varphi(q)) - w(q)\big) \\
&= \sum_{q \in Q_j} \big(w(\varphi(q)) - w(q)\big) - \sum_{q \in Q_i} \big(w(\varphi(q)) - w(q)\big) \\
&= \delta(Q_j) - \delta(Q_i) = 0
\end{aligned}$$

which implies that the weight of the spanning tree $(Q \setminus Q') \cup \varphi(Q')$ is equal to $w(Q)$ contradicting $\sigma(B, Q) = 1$. Hence, there are at least $n-1$ spanning trees with distinct weight such that each of them is smaller than $w(Q)$ implying $\operatorname{ord}(B, Q) \geq n$ which proves the theorem. □

Observe that because of Lemma 3.11 we only counted spanning trees T of a bispanning graph $B = (V, P, Q)$ such that the remaining edges $(P \cup Q) \setminus T$ also form a spanning tree. There are strong indications that these spanning trees are sufficient to prove Conjecture 6. A question which arises now is how to distinguish bispanning graphs whose cycle matroid is strongly base orderable from bispanning graphs that do not have this property? The following theorem gives a first answer to this question.

Theorem 3.13. *Transversal matroids are strongly base orderable.*

This proof is a simplification of the proof given in [Gam99].

Proof. Given two arbitrary bases X and Y of a transversal matroid M on a ground set E, we construct a strongly base orderable bijection $\varphi\colon X \to Y$. Without loss of generality, we assume $X \cap Y = \emptyset$ since otherwise we set $\varphi(x) := x$ for all $x \in X \cap Y$ to obtain a strongly base orderable bijection.

The construction of φ is starting from the bipartite graph representation $G(E, \mathcal{A})$ for an arbitrary presentation $\mathcal{A}$ of M. Let $M_X\colon X \to \mathcal{A}$ and $M_Y\colon Y \to \mathcal{A}$ be two matchings in $G(E, \mathcal{A})$ of cardinality $|X| = |Y|$. Both matchings must exist since M is a transversal matroid.

Let $x \in X$ be an arbitrary element of X and $A \in \mathcal{A}$ such that x is matched into A by M_X. We observe that there must be an element $y \in Y$ such that y is matched by M_Y into A. Otherwise $Y \cup \{x\}$ is a transversal in contradiction to Y being a base. We define $\varphi(x) := y$. The sets $(X \setminus \{x\}) \cup \{\varphi(x)\}$ and $(Y \setminus \{\varphi(x)\}) \cup \{x\}$ are transversals. Moreover, exchanging x and $\varphi(x)$ does not have an effect on any other exchanges. Thus, φ is a strongly base orderable bijection. □

The converse of Theorem 3.13, i.e., a matroid is transversal if it is strongly base orderable, is not true. To show a counterexample, we need the following theorem, which proposes a property a graph has to achieve such that its cycle matroid is transversal. For a proof, which requires a deeper look into matroid theory, we refer the reader to [Bon72] and [Wel76].

Theorem 3.14. *Let $G = (V, E)$ be a finite graph. Then its cycle matroid $M = (E, \mathcal{I})$ is transversal if and only if K_4 and C_k^2 for all $k > 2$ are no topological minors of G.*

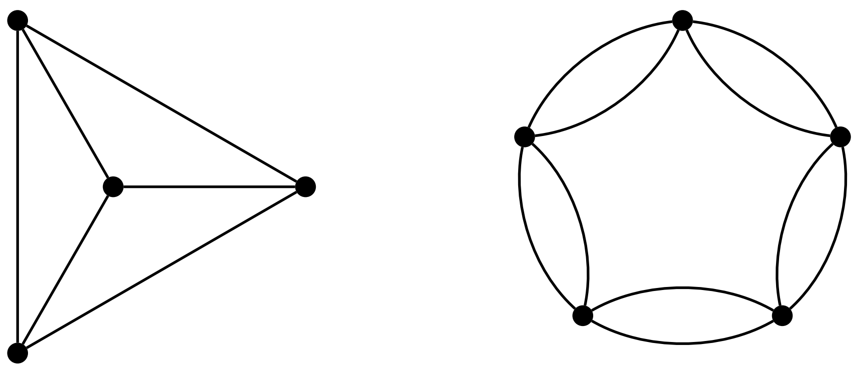


Figure 3.4: K_4 and C_5^2.

Now, we consider the graph given in Figure 3.5. Obviously, this graph can be obtained by two edge-subdivisions of any pair of multiple edges of the C_3^2. Hence, the cycle matroid of this graph is not transversal by Theorem 3.14. But there exists, for each two spanning trees T and T' of this graph, a bijection $\varphi\colon T \to T'$ such that for each subset $E \subseteq T$, the set $(T \setminus E) \cup \varphi(E)$ is a spanning tree, too. The bijection φ can be constructed as follows. Without loss of generality, we assume $T \cap T' = \emptyset$ since

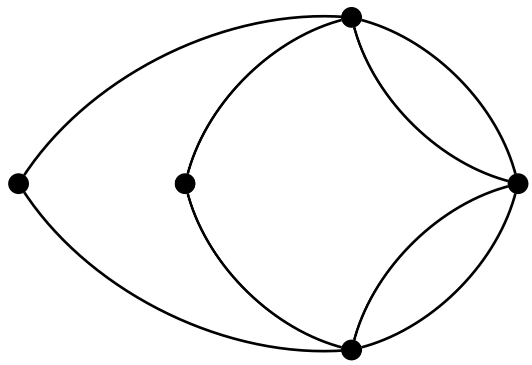


Figure 3.5: The cycle matroid is strongly base orderable but not transversal.

otherwise, we set $\varphi(e) = e$ for all $e \in T \cap T'$. Now, we consider the bispanning graph $B = (V, T, T')$. This graph has at least one vertex v of degree $\deg(v) = 2$. Let $e \in T$ and $f \in T'$ be the corresponding edges incident to v. We set $\varphi(e) = f$, cut off both leaves (and the vertex v) and recursively repeat this procedure until the graph is exhausted. For a deeper view regarding the correctness of this approach, we refer the reader to Chapters 4 and 5.

In the following, we will show that there exists a better characterization of graphs whose cycle matroid is strongly base orderable. Actually, we will see that it suffices to look for a (topological) minor isomorphic to K_4. First, we consider the following definition. A matroid M is called *base orderable* if for each two bases B and B' of M, there exists a bijection $\varphi\colon B \to B'$ such that for all $x \in B$, the sets $(B \setminus \{x\}) \cup \{\varphi(x)\}$ and $(B' \setminus \{\varphi(x)\} \cup \{x\}$ are bases of M. Obviously, if M is strongly base orderable then M is also base orderable. For some time, it was unknown whether there exists a matroid which is base orderable but not strongly base orderable. Ingleton was the first who found an example of such a matroid [Ing75]. Bondy gave the following characterization for a base orderable cycle matroid of a graph.

Theorem 3.15. *Let G be a graph. Then, its cycle matroid is base orderable if and only if G contains no (topological) minor isomorphic to K_4.*

When speaking about the cycle matroid of a graph, both definitions are equivalent as we will see in the following. The cycle matroid of a graph is also known to be a binary matroid. A matroid is a binary matroid if it can be represented over $GF(2)$, the Galois field of two elements. For a graph $G = (V, E)$, we take the vertex-edge-incidence matrix and delete an arbitrary row. Then, any set of linear independent columns is defined to be an independent set. These sets correspond to the forests of G. Keijsper, Pendaving, and Schrijver proved that for binary matroids, the property of being base orderable is equivalent to being strongly base orderable [KPS00]. Hence, we obtain the following corollary.

Corollary 3.16. *The cycle matroid of a graph $G = (V, E)$ is strongly base orderable if and only if G has no (topological) minor isomorphic to K_4.*

A related exchange property was studied by Gabow [Gab76]. Given a matroid M and a base B of M, he defined a sequence of ordered pairs of sets X_i, X_i', $i = 1, \ldots, m$, to be a *serial B-exchange* if $B_0 = B$ and $B_i = (B_{i-1} \setminus X_i) \cup X_i'$ are bases of M and $X_i \subset B_{i-1}$ holds for all $i = 1, \ldots m$. The ordered pair X, X' is called a *strong serial B-exchange* if there exists a bijection $\varphi \colon X \to X'$ such that for any ordering of the elements in X, the corresponding sequence of pairs is a serial B-exchange, i.e., the exchange of X can be executed in any order. Gabow proved the following decomposition theorem.

Theorem 3.17. *Let X, X' be a symmetric exchange with respect to two bases B, B' of a matroid. Then, the sets X and X' can be partitioned*

$$X = \bigcup_{i=1}^{m} Y_i \qquad \text{and} \qquad X' = \bigcup_{i=1}^{m} Y_i'$$

such that

(i) the sequence Y_i, Y_i', $i = 1, \ldots, m$, is a serial B-exchange,

(ii) the sequence Y_i', Y_i, $i = 1, \ldots, m$, is a serial B'-exchange, and

(iii) the exchanges made in (i) and (ii), Y_i, Y_i' and Y_i', Y_i, are strong serial exchanges.

We obtain a special case of the theorem if $X = B$ and $X' = B'$. Here, it is interesting to ask whether or not it can hold $m = 1$, which corresponds to a strongly base ordering. Analogously, to our studies in this chapter, Gabow discovered that this case cannot always be achieved. Again, the complete graph on four vertices K_4 was given as a counterexample.

Chapter 4

Partitioning Bispanning Graphs

4.1 Introduction

In this chapter, we refine Conjecture 6 by merging the different findings which we discussed in Chapter 3. In Section 3.2, we have seen that this conjecture holds under the assumption that the weight function is also required to satisfy $\sigma(B, P) = 1$. The weight of spanning tree P is generally not unique even if the number of vertices is small. Figure 4.1 shows an example of the complete graph on four vertices K_4 which is the

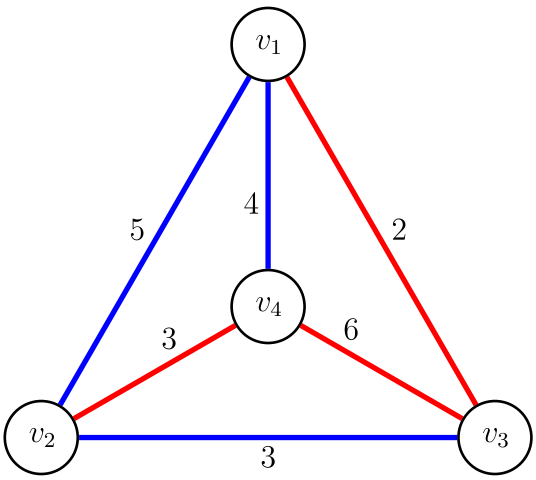


Figure 4.1: Example of a weighted bispanning graph $B = (V, P, Q, w)$ such that $\sigma(B, Q) = 1$ and $\sigma(B, P) = 4$.

smallest bispanning graph $B = (V, P, Q)$ without multiple edges (aside from the trivial bispanning graph which is a single vertex). Recall that the edges of Q are blue-colored and the edges of P are red-colored, respectively. Moreover, we observe that there is (up to isomorphism) only one partition of the K_4 into two distinct spanning trees. The spanning tree weights in Figure 4.1 are $w(P) = 11$ and $w(Q) = 12$, thus, it holds that $\text{ord}(B, P) < \text{ord}(B, Q)$. One can easily check by an exhaustive enumeration of all spanning trees that $\sigma(B, Q) = 1$ and $\sigma(B, P) = 4$ hold since

$$w(P) = 11 = \underbrace{2+4+5}_{(v_1, V \setminus \{v_1\})} = \underbrace{3+3+5}_{(v_2, V \setminus \{v_2\})} = \underbrace{2+3+6}_{(v_3, V \setminus \{v_3\})}$$

where $(v_i, V \setminus \{v_i\})$ denotes the edges joining v_i with a vertex in $V \setminus \{v_i\}$, $i \in \{1, 2, 3\}$. In this case, the main observation is that given a spanning tree $T \neq P$ with $w(T) = w(P)$, the remaining edges $E \setminus T$ contain a cycle. In Figure 4.1, the remaining edges cannot form a spanning tree since each spanning tree with weight $w(P)$ contains all edges of a cut. To avoid the problem of $\sigma(B, P) > 1$, we introduce the concept of partitioning bispanning graphs into spanning trees which was already indicated in Section 3.3. This approach leads to a somewhat stronger conjecture compared to Conjecture 6.

Definition 4.1. *Let $B = (V, P, Q)$ be a bispanning graph. A spanning tree T of B is called a partition spanning tree if its complement $E \setminus T$ is a spanning tree, too.*

Let $B = (V, P, Q)$ be a bispanning graph. We denote by $\mathcal{T}'(B)$ the set of all partition spanning trees of B. Given a weight function $w \colon (P \cup Q) \to \mathbb{R}$ we denote by $\mathcal{W}'(B)$ the set of different weights of all partition spanning trees of B and by $\mathcal{W}'_i(B)$ the ith smallest element of $\mathcal{W}'(B)$. Moreover, $\mathcal{T}'_i(B)$ is the set of partition spanning trees T where $w(T) = \mathcal{W}'_i(B)$. We define the order $\mathrm{ord}'(B, T)$ of a partition spanning tree T with respect to B as the number $i \in \mathbb{N}$ such that $T \in \mathcal{T}'_i(B)$. The number of partition spanning trees with weight $w(T)$ is denoted by $\sigma'(B, T)$, that is, $\sigma'(B, T) = |\mathcal{T}'_{\mathrm{ord}'(B,T)}(B)|$.

Conjecture 7. *Let $B = (V, P, Q, w)$ be a weighted bispanning graph such that $\mathrm{ord}(B, P) < \mathrm{ord}(B, Q)$ and $\sigma(B, Q) = 1$. Then, it holds that $\mathrm{ord}'(B, Q) \geq n$.*

If Conjecture 7 holds then it immediately implies Conjecture 6 since we have

$$\mathrm{ord}'(B, T) \leq \mathrm{ord}(B, T) \tag{4.1}$$

for all partition spanning trees T of B. Using the 'symmetric subset exchange axiom' (Lemma 3.1) for spanning trees, we can give a lower bound on the number of partition spanning trees with (not necessarily distinct) weights less than $w(Q)$.

Proposition 4.2. *Let $B = (V, P, Q, w)$ be a weighted bispanning graph such that $\mathrm{ord}(B, P) < \mathrm{ord}(B, Q)$ and $\sigma(B, Q) = 1$. Then there are at least 2^{n-2} weighted partition spanning trees T in B such that $w(T) < w(Q)$.*

Proof. There are 2^{n-1} subsets P' of P. According to Lemma 3.1, there is a subset Q' of Q for each P' such that $(P \setminus P') \cup Q'$ as well as $(Q \setminus Q') \cup P'$ are partition spanning trees. Clearly, the weight of at least one of both sets is less than $w(Q)$. Because of symmetry, we have to divide by two and the claim follows. $\square$

4.2 Strictly 2-Edge-Connected Bispanning Graphs

In this section, we consider strictly 2-edge-connected bispanning graphs and show that each of these bispanning graphs can be reduced to some 3-edge-connected bispanning graph under the assumption that it is only necessary to count partition spanning trees. Remember that $B[q, p]$ denotes the bispanning graph which is obtained from B by contracting q and discarding p, i.e., $B[q, p] = (B \setminus p)/q$.

Theorem 4.3. *Let $B = (V, P, Q, w)$ be a weighted bispanning graph with edge-connectivity $\lambda(B) = 2$, $\operatorname{ord}(B, P) < \operatorname{ord}(B, Q)$, and $\sigma(B, Q) = 1$. Then, there are two edges $p \in P$ and $q \in Q$ such that $\operatorname{ord}'(B[q, p], Q[q, p]) < \operatorname{ord}'(B, Q)$.*

Proof. Since $\lambda(B) = 2$, there exists a cut $(V', V \setminus V')$ in B with exactly two edges between V' and $V \setminus V'$. Clearly, one of these edges belongs to P and the other one belongs to Q since otherwise either P or Q is not a spanning tree. We denote by p the edge which belongs to P and by q the edge which belongs to Q, respectively. Now, we consider the bispanning graph $B' = B[q, p]$. In Figure 4.2, there is an illustration of this transformation where p is removed and the edge q connecting u and v is contracted to a new vertex $\overline{uv}$.

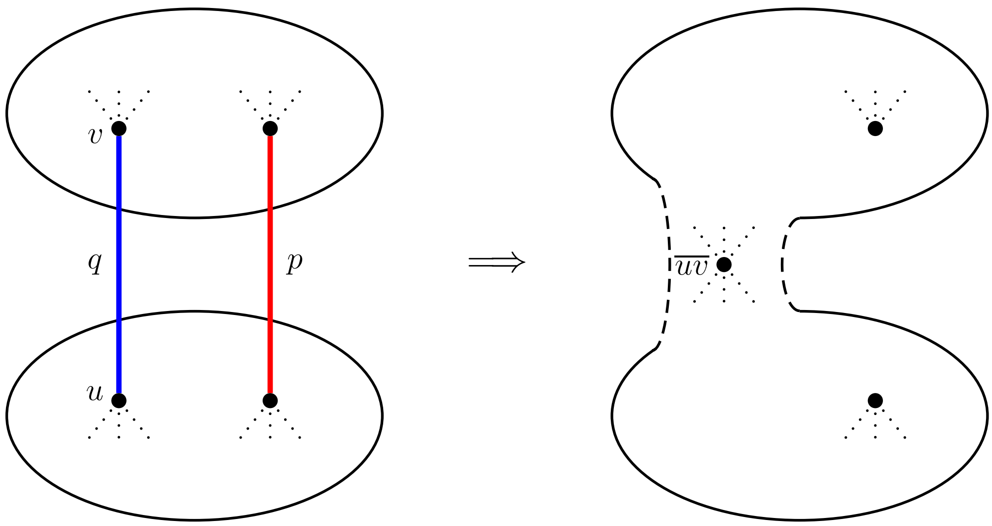


Figure 4.2: Transforming strictly 2-edge-connected bispanning graphs.

As already seen in Section 3.2 (cf. Lemma 3.2), each spanning tree of $B' = B[q, p]$ can be combined with the edge q yielding a spanning tree of B. If we consider a cut consisting of two edges p and q, we can extend this fact to obtain a stronger proposition. Now, each partition spanning tree of B' can be combined either with p or with q yielding a partition spanning tree of B. Depending on the weights of p and q, the sequence of different partition spanning tree weights of B' in increasing order is either

$$(T'_1, \ldots, T'_{\alpha-1}, T'_\alpha = P', T'_{\alpha+1}, \ldots, T'_{\beta-1}, T'_\beta = Q', T'_{\beta+1}, \ldots, T'_\gamma) \tag{4.2}$$

or

$$(T'_1, \ldots, T'_{\alpha-1}, T'_\alpha = Q', T'_{\alpha+1}, \ldots, T'_{\beta-1}, T'_\beta = P', T'_{\beta+1}, \ldots, T'_\gamma) \tag{4.3}$$

for appropriate values of α and β. For the sake of readability, we associate in (4.2) and (4.3) with a tree T also its weight $w(T)$. Because of the symmetric properties of partition spanning trees, the value of γ is clearly defined to be $\gamma = \alpha + \beta - 1$. Furthermore, for each $1 \leq i \leq \alpha + \beta - 1$, we have $w(T'_i) + w(T'_{\alpha+\beta-i}) = w(B')$ where $w(B')$ is the total sum of weights of edges in B'.

If (4.2) holds, we combine each partition spanning tree of B' with the edge q resulting in $\beta \geq n-1$ partition spanning trees with distinct weights where each of them is smaller than or equal to $w(Q)$. Because P is the only partition spanning tree of weight $w(P)$, none of these weights can map into $w(P)$. Hence, there are $\beta+1 \geq n$ partition spanning tree in B implying $\operatorname{ord}'(B', Q') < \operatorname{ord}'(B, Q)$.

In the second case, i.e., (4.3) holds, we combine each partition spanning tree with the edge p. Since $w(P' \cup \{p\}) < w(Q' \cup \{q\})$, we arrive at $\operatorname{ord}'(B', Q') < \operatorname{ord}'(B, Q)$ and the claim follows. Both cases are illustrated in Figure 4.3 and Figure 4.4. $\square$

As a consequence of Theorem 4.3, it suffices to consider 3-edge-connected bispanning graphs to prove Conjecture 7.

4.3 Decomposition of Bispanning Graphs with Cut Vertices

In the previous section, we have seen that we can turn our attention to 3-edge-connected bispanning graphs. Now, we want to show that it is also sufficient to consider only 2-vertex-connected bispanning graphs, i.e., graphs that do not contain any cut vertex. In the following theorem, we assume that the given bispanning graph B contains at least one cut vertex. If the number of cut vertices is greater than one, it is possible to recursively apply this theorem to its 2-connected components.

Theorem 4.4. *Let $B = (V, P, Q, w)$ be a weighted bispanning graph with a cut vertex $v \in V$ and such that $\operatorname{ord}(B, P) < \operatorname{ord}(B, Q)$ and $\sigma(B, Q) = 1$ holds. Let $B_1 = (V_1, P_1, Q_1, w)$ and $B_2 = (V_2, P_2, Q_2, w)$ be the two bispanning subgraphs which are connected through v. Then B_1 and B_2 are weighted bispanning graphs such that Q_1 and Q_2 have unique weights. Furthermore, if Conjecture 7 holds for both of them, Conjecture 7 holds for B.*

In the proof of Theorem 4.3, we combined partition spanning trees with the cut edges p and q. Now, we have to combine partition spanning trees of B_1 with partition spanning tree of B_2. Since this is slightly more difficult, we take a look at the following two lemmas because they will simplify the proof of Theorem 4.4.

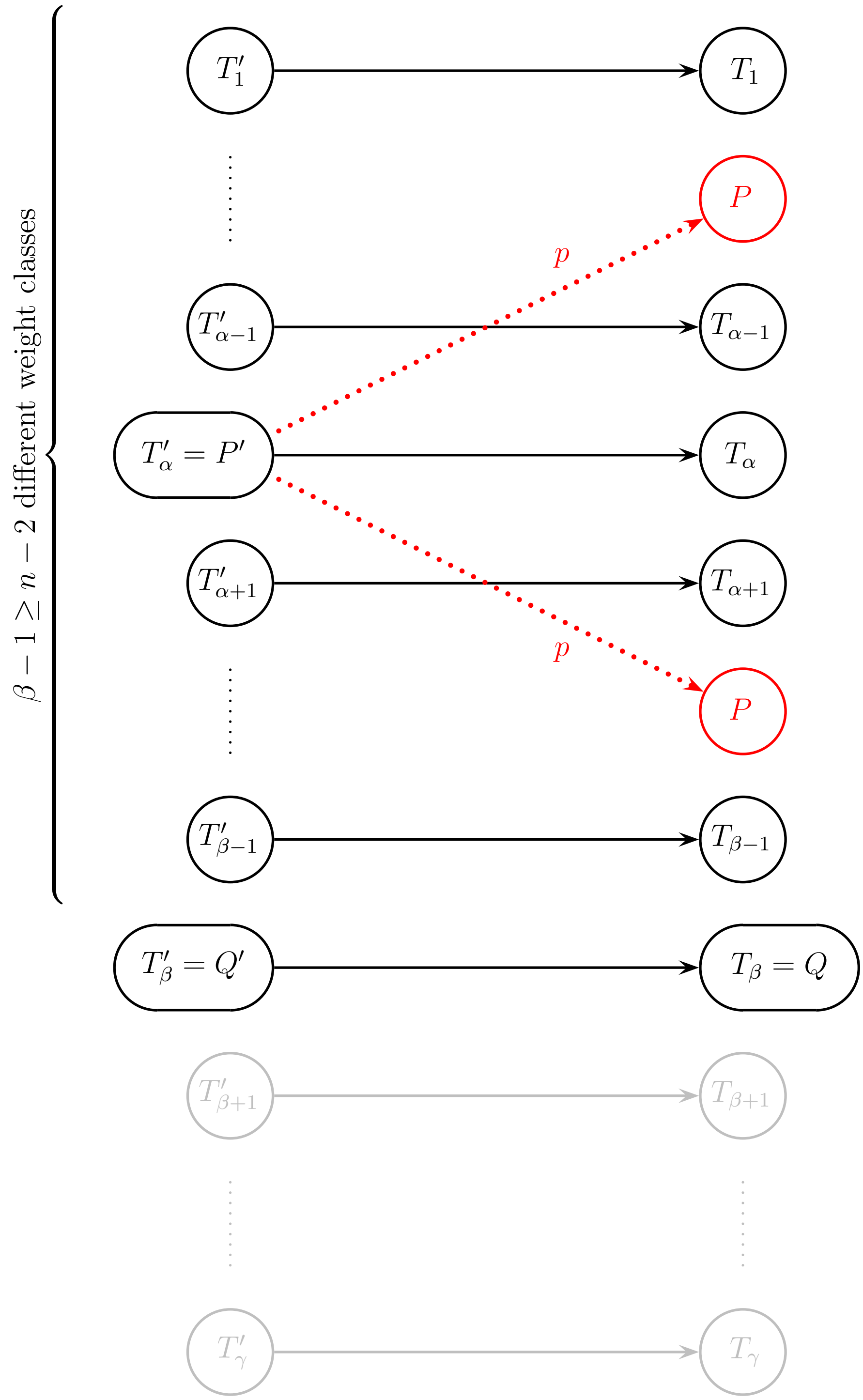


Figure 4.3: Each partition spanning tree of $B' = B[q, p]$ can be combined with the edge q (solid lines). Depending on whether $w(q) < w(p)$ or $w(p) < w(q)$ (together with $w(P') < w(Q')$) holds, we get either the upper partition spanning tree P or the lower one.

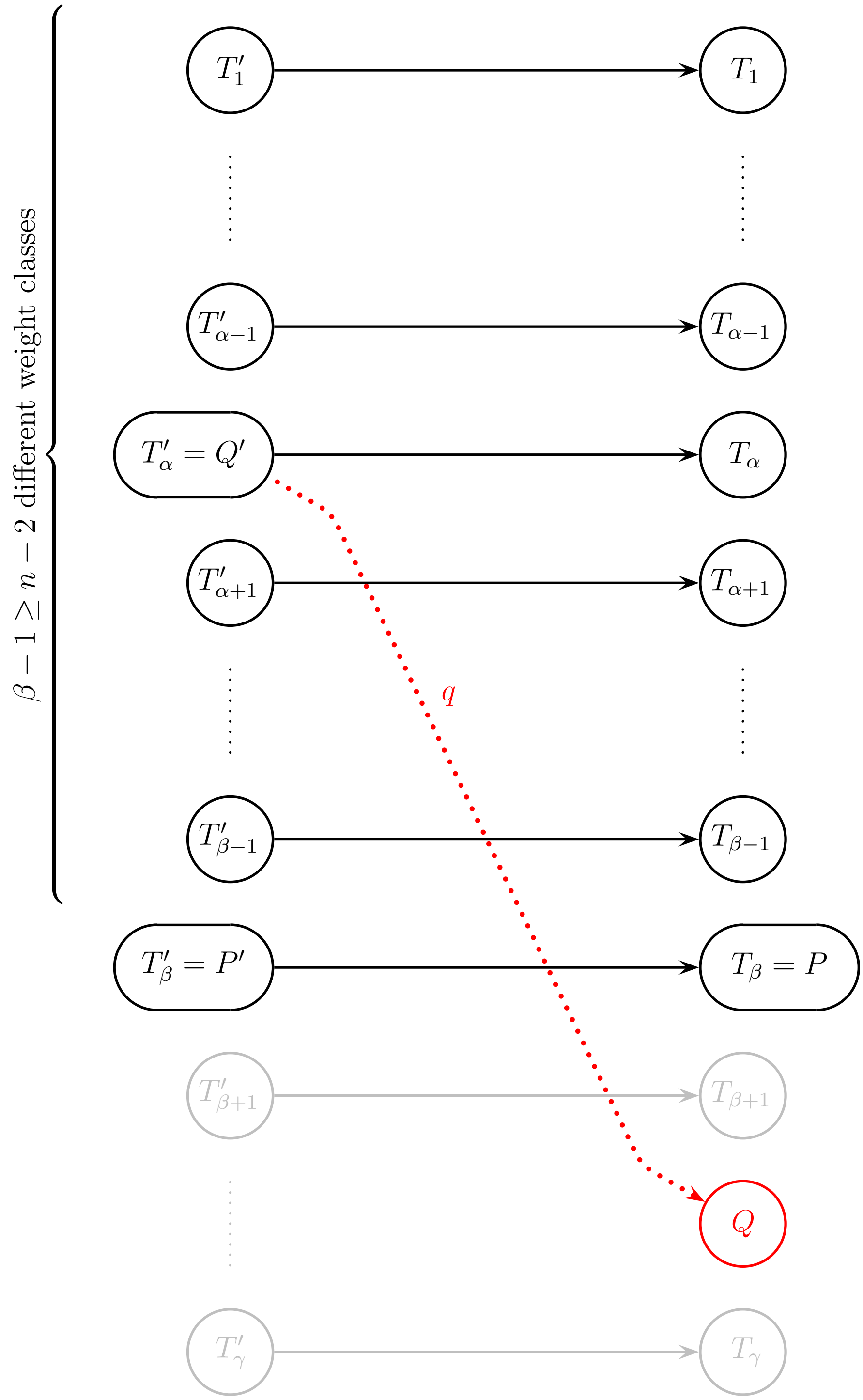


Figure 4.4: Constructing new classes of partition spanning trees by adding the edge p (solid lines) to the classes of $B[q,p]$.

Lemma 4.5. *Let $X = (x_1, x_2, \ldots, x_\beta)$ and $Y = (y_1, y_2, \ldots, y_\nu)$ be sequences of numbers such that $x_i < x_j$ for all $1 \leq i < j \leq \beta$ and $y_k < y_\ell$ for all $1 \leq k < \ell \leq \nu$. Let $S = \{x_i + y_k \mid 1 \leq i \leq \beta \text{ and } 1 \leq k \leq \nu\}$ be the set of all possible sums of two elements $x \in X$ and $y \in Y$. Then, there are at least $\beta+\nu-2$ distinct $s \in S$ such that $s < x_\beta + y_\nu$.*

Proof. Given two strictly increasing sequences of numbers $X = (x_1, x_2, \ldots, x_\beta)$ and $Y = (y_1, y_2, \ldots, y_\nu)$, the following chain consists of $\beta + \nu - 2$ distinct sums which are strictly smaller than $x_\beta + y_\nu$:

$$\underbrace{(x_1 + y_1) < (x_1 + y_2) < \ldots < (x_1 + y_{\nu-1})}_{\nu-1 \text{ pairs}} <$$
$$< \underbrace{(x_1 + y_\nu) < (x_2 + y_\nu) < \ldots < (x_{\beta-1} + y_\nu)}_{\beta-1 \text{ pairs}} < (x_\beta + y_\nu)$$

This proves the lemma. □

The next lemma is an extension of Lemma 4.5 containing further properties.

Lemma 4.6. *Let $X = (x_1, \ldots, x_\alpha, \ldots, x_\beta)$ and $Y = (y_1, \ldots, y_\mu, \ldots, y_\nu)$ be sequences of numbers such that $\alpha < \beta$, $\mu < \nu$, $x_i < x_j$ for all $1 \leq i < j \leq \beta$, $y_k < y_\ell$ for all $1 \leq k < \ell \leq \nu$, and such that the following restrictions are satisfied:*

1. *Let E_X, E_Y, F_X, and F_Y defined as*

$$\begin{aligned} E_X &= \{x_\beta - x_i \mid 1 \leq i < \beta\} \\ E_Y &= \{y_\nu - y_k \mid 1 \leq k < \nu\} \\ F_X &= \{x_j - x_\alpha \mid \alpha < j < \beta\} \\ F_Y &= \{y_\ell - y_\mu \mid \mu < \ell < \nu\} , \end{aligned}$$

   we assume X and Y satisfy $F_X \subseteq E_X$, $F_Y \subseteq E_Y$, and $F_X \cap F_Y = \emptyset$.

2. *$x_\beta + y_\mu < x_\alpha + y_\nu$.*

3. *$x_i + y_k = x_\alpha + y_\nu$ if and only if $i = \alpha$ and $k = \nu$.*

Then $S = \{x_i + y_k \mid 1 \leq i \leq \beta \text{ and } 1 \leq k \leq \nu\}$, which is the set of all possible sums of two elements $x \in X$ and $y \in Y$, consists of at least $\beta + \nu - 2$ distinct elements $s \in S$ such that $s < x_\alpha + y_\nu$.

Proof. We consider the following chain of pairwise different sums

$$\underbrace{(x_1 + y_1) < (x_1 + y_2) < \ldots < (x_1 + y_{\mu-1})}_{\mu-1 \text{ pairs}} <$$
$$< \underbrace{(x_1 + y_\mu) < (x_2 + y_\mu) < \ldots < (x_\alpha + y_\mu) < \ldots < (x_\beta + y_\mu)}_{\beta \text{ pairs}} .$$

Obviously, all of these sums are less than $x_\alpha + y_\nu$, that is, we have already found $\beta+\mu-1$ distinct sums and we have to show that there are $\nu - \mu - 1$ further distinct pairs with this property. To this end, we consider the sums that are formed by x_α and y_k for $\mu < k < \nu$. Clearly, these sums are distinct where each of them is greater than $x_\alpha + y_\mu$ and smaller than $x_\alpha + y_\nu$. By condition (3) in Lemma 4.6, they can only conflict with some pair $x_i + y_\mu$, $\alpha < i < \beta$, in the chain given above. Suppose there is a collision with some pair, that is, there exists some $\alpha < i < \beta$ and $\mu < k < \nu$ such that $x_\alpha + y_k = x_i + y_\mu \iff y_k - y_\mu = x_i - x_\alpha$. Then, we obtain a contradiction to our assumption (condition (1) in Lemma 4.6) $F_X \cap F_Y = \emptyset$ since $x_i - x_\alpha \in F_X$ and $y_k - y_\mu \in F_Y$ for these values of i and k. Hence, there are at least $\beta + \mu - 1 + \nu - \mu - 1 = \beta + \nu - 2$ distinct elements $s \in S$ such that $s < x_\alpha + y_\nu$ proving the lemma. □

Proof of Theorem 4.4. Given the bispanning graph $B = (V, P, Q)$ and a cut vertex v in B, the graph decomposes into two bispanning subgraphs, $B_1 = (V_1, P_1, Q_1)$ and $B_2 = (V_2, P_2, Q_2)$, which are connected by v (see Figure 4.5 for an illustration). Clearly,

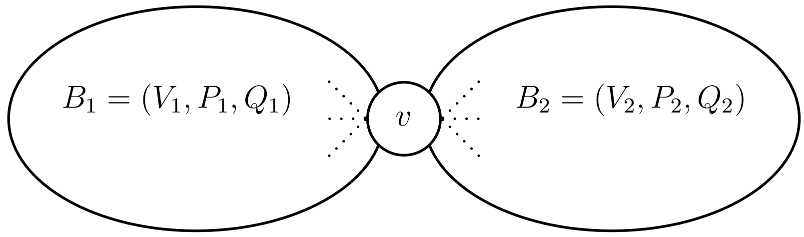


Figure 4.5: Two bispanning graphs joined by a cut vertex.

in both subgraphs, the remaining part of the spanning tree Q has unique weight, i.e., it holds that $\sigma(B_1, Q_1) = 1$ and $\sigma(B_2, Q_2) = 1$ since otherwise we easily get a contradiction to $\sigma(B, Q) = 1$ by constructing another spanning tree $T \neq Q$ with weight $w(T) = w(Q)$. Another fact is that each spanning tree of B_1 together with each spanning tree of B_2 forms a spanning tree of B. Moreover, a pair of *partition* spanning trees of B_1 and B_2 forms a *partition* spanning tree of B.

Since $w(P_1) + w(P_2) < w(Q_1) + w(Q_2)$ not both inequations, $w(P_1) > w(Q_1)$ *and* $w(P_2) > w(Q_2)$, can be true. Hence, without loss of generality, we assume $w(P_2) < w(Q_2)$. Depending on whether $w(P_1) < w(Q_1)$ or $w(P_1) > w(Q_1)$ holds, we distinguish between

$$\operatorname{ord}'(B_1, P_1) < \operatorname{ord}'(B_1, Q_1) \qquad \text{and} \qquad \operatorname{ord}'(B_2, P_2) < \operatorname{ord}'(B_2, Q_2) \tag{4.4}$$

and

$$\operatorname{ord}'(B_1, P_1) > \operatorname{ord}'(B_1, Q_1) \qquad \text{and} \qquad \operatorname{ord}'(B_2, P_2) < \operatorname{ord}'(B_2, Q_2)\,. \tag{4.5}$$

On the one hand, if (4.4) holds, the ordered sequences of all partition spanning tree weights of B_1 and B_2 are

$$(T_1^{(1)}, \ldots, T_{\alpha-1}^{(1)}, T_\alpha^{(1)} = P_1, T_{\alpha+1}^{(1)}, \ldots, T_{\beta-1}^{(1)}, T_\beta^{(1)} = Q_1, T_{\beta+1}^{(1)}, \ldots, T_\gamma^{(1)}) \tag{4.6}$$

and

$$(T_1^{(2)}, \ldots, T_{\mu-1}^{(2)}, T_\mu^{(2)} = P_2, T_{\mu+1}^{(2)}, \ldots, T_{\nu-1}^{(2)}, T_\nu^{(2)} = Q_2, T_{\nu+1}^{(2)}, \ldots, T_\kappa^{(2)}) \ . \tag{4.7}$$

By assumption, Conjecture 7 holds for both bispanning subgraphs, that is, we have $\operatorname{ord}'(B_1, Q_1) = \beta \geq n_1$ and $\operatorname{ord}'(B_2, Q_2) = \nu \geq n_2$ with $n_1 := |V_1|$ and $n_2 := |V_2|$. The number of vertices in B is $|V| = n = n_1 + n_2 - 1$. Thus, we have to construct $n_1 + n_2 - 2$ partition spanning trees with distinct weights where each weight is smaller than $w(Q)$. To this end, we apply Lemma 4.5 with (the sequences of tree weights)

$$X = (T_1^{(1)}, T_2^{(1)}, \ldots, T_\beta^{(1)} = Q_1)$$

and

$$Y = (T_1^{(2)}, T_2^{(2)}, \ldots, T_\nu^{(2)} = Q_2)$$

obtaining $\beta + \nu - 2 \geq n_1 + n_2 - 2$ partition spanning trees with distinct weights strictly less than $w(Q)$. All combinations according to Lemma 4.5 are illustrated in Figure 4.6 where it is easy to see that all of them lead to partition spanning trees of different weights since there are no crossing lines.

On the other hand, we suppose (4.5) holds. Again, we consider the ordered sequences of all partition spanning tree weights of B_1 and B_2 which are

$$(T_1^{(1)}, \ldots, T_{\alpha-1}^{(1)}, T_\alpha^{(1)} = Q_1, T_{\alpha+1}^{(1)}, \ldots, T_{\beta-1}^{(1)}, T_\beta^{(1)} = P_1, T_{\beta+1}^{(1)}, \ldots, T_\gamma^{(1)}) \tag{4.8}$$

and

$$(T_1^{(2)}, \ldots, T_{\mu-1}^{(2)}, T_\mu^{(2)} = P_2, T_{\mu+1}^{(2)}, \ldots, T_{\nu-1}^{(2)}, T_\nu^{(2)} = Q_2, T_{\nu+1}^{(2)}, \ldots, T_\kappa^{(2)}) \ . \tag{4.9}$$

Accordingly to the previous case, we have $\sigma(B_1, Q_1) = \sigma(B_2, Q_2) = 1$. Assuming Conjecture 7 holds for B_1 and B_2, we have $\operatorname{ord}'(B_2, Q_2) = \nu \geq n_2$ with $n_2 := |V_2|$ and $\operatorname{ord}'(B_1, P_1) = \beta \geq n_1$ with $n_1 := |V_1|$ where the latter lower bound follows because of the symmetric properties of partition spanning tree weights.

In the previous case, we applied Lemma 4.5 where it was not difficult to construct $\beta + \nu - 2$ distinct pairs of partition spanning trees. Because the relation between Q_1 and P_1 in (4.8) has changed, it is not as easy as in the previous case. However, together with Lemma 4.6, we are able to prove the same number of partition spanning trees. To this end, let

$$X = (T_1^{(1)}, \ldots, T_\alpha^{(1)}, \ldots, T_\beta^{(1)})$$

and

$$Y = (T_1^{(2)}, \ldots, T_\mu^{(2)}, \ldots, T_\nu^{(2)})$$

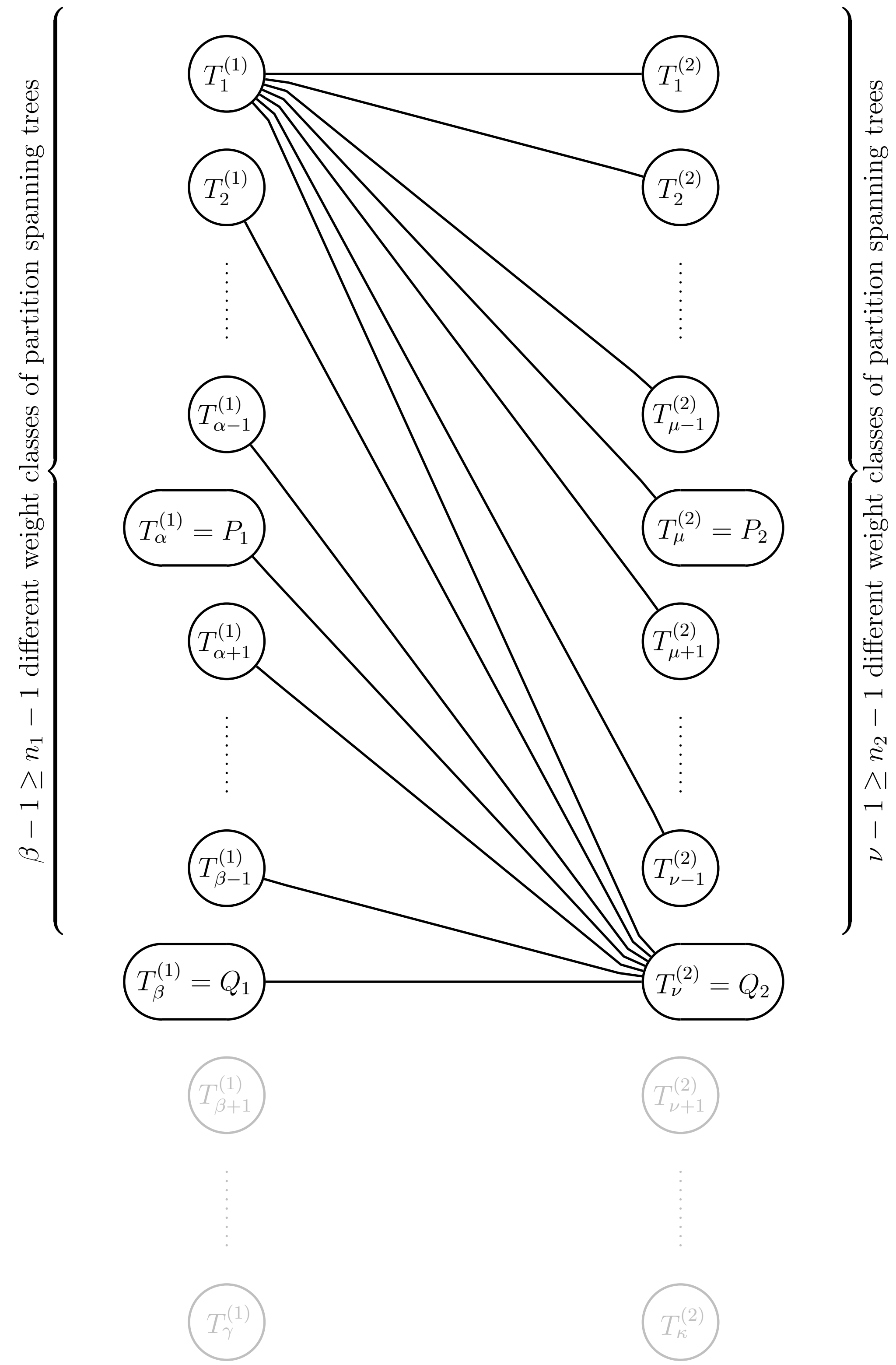


Figure 4.6: Combinations of partition spanning trees if $w(P_1) < w(Q_1)$ and $w(P_2) < w(Q_2)$.

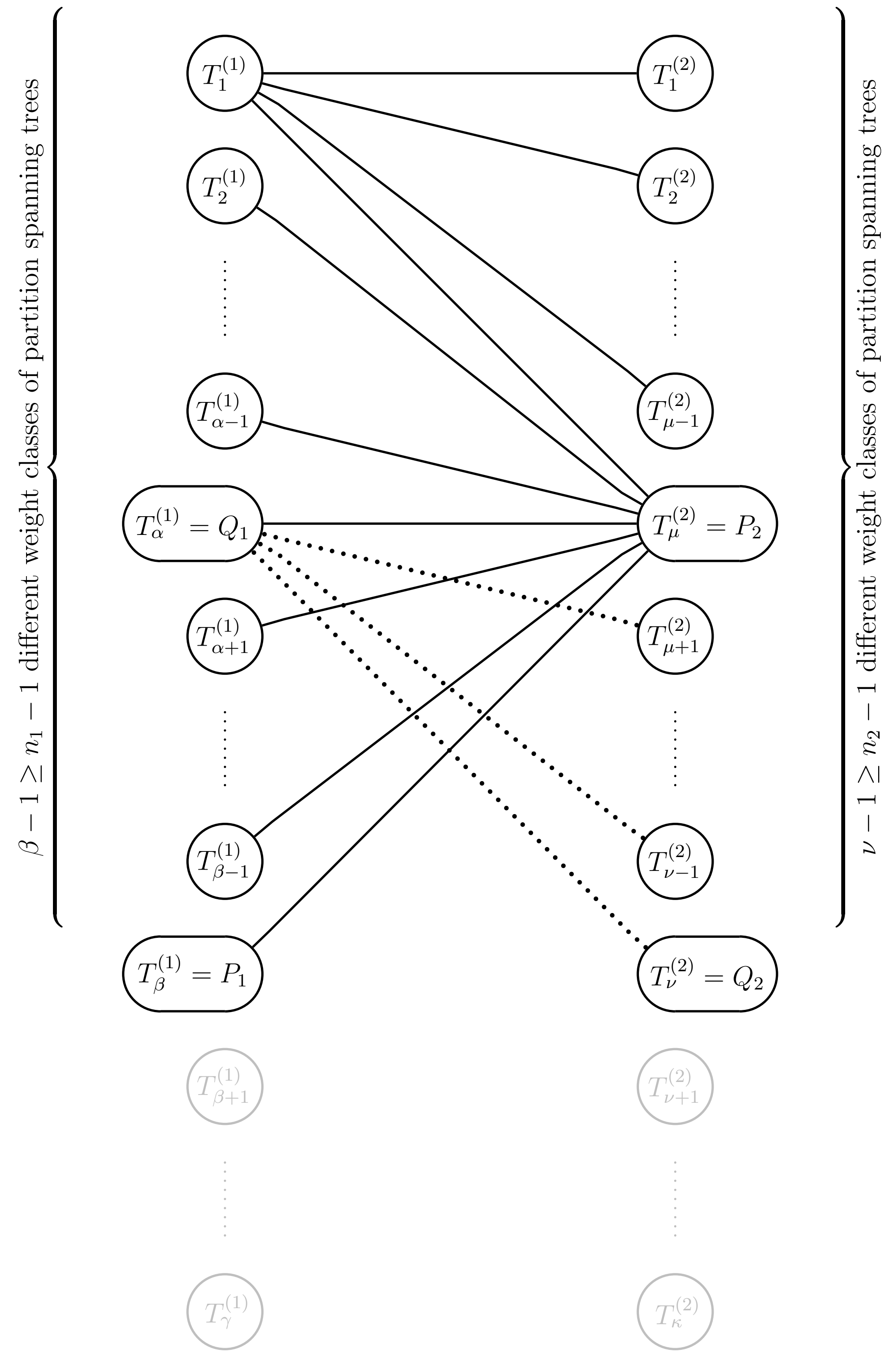


Figure 4.7: Combinations of partition spanning trees if $w(P_1) > w(Q_1)$ and $w(P_2) < w(Q_2)$.

with $T^{(1)}_\alpha = Q_1$, $T^{(1)}_\beta = P_1$, $T^{(2)}_\mu = P_2$, and $T^{(2)}_\nu = Q_2$. Furthermore, let E_X, E_Y, F_X, and F_Y be defined as

$$\begin{aligned}
E_X &= \{T^{(1)}_\beta - T^{(1)}_i \mid 1 \le i < \beta\} \\
E_Y &= \{T^{(2)}_\nu - T^{(2)}_k \mid 1 \le k < \nu\} \\
F_X &= \{T^{(1)}_j - T^{(1)}_\alpha \mid \alpha < j < \beta\} \\
F_Y &= \{T^{(2)}_\ell - T^{(2)}_\mu \mid \mu < \ell < \nu\} \ .
\end{aligned}$$

Remember that the partition spanning tree weights satisfy a symmetric property which can be formulated as

$$T_i - T_\alpha = T_\beta - T_{\alpha+\beta-i} \tag{4.10}$$

for all $\alpha < i < \beta$ with respect to the sequence X and

$$T_k - T_\mu = T_\nu - T_{\mu+\nu-k} \tag{4.11}$$

for all $\mu < k < \nu$ with respect to the sequence Y. Thus, X and Y satisfy the three conditions of Lemma 4.6:

1. Because of (4.10) and (4.11), it holds that $F_X \subseteq E_X$ and $F_Y \subseteq F_Y$. Moreover, we have $F_X \cap F_Y = \emptyset$ since otherwise there exist indices $\alpha < j < \beta$ and $\mu < \ell < \nu$ such that $T^{(1)}_j - T^{(1)}_\alpha = T^{(2)}_\ell - T^{(2)}_\mu = T^{(2)}_\nu - T^{(2)}_{\mu+\nu-k} \iff T^{(1)}_\alpha + T^{(2)}_\nu = Q_1 + Q_2 = T^{(1)}_j + T^{(2)}_{\mu+\nu-k}$ implying a contradiction to $\sigma(B, Q) = 1$.

2. Clearly, we have $T^{(1)}_\beta + T^{(2)}_\mu = P_1 + P_2 = P < Q = Q_1 + Q_2 = T^{(1)}_\alpha + T^{(2)}_\nu$.

3. This condition holds because of $\sigma(B, Q) = 1$.

Hence, we arrive at $\mathrm{ord}'(B, Q) = \beta + \nu - 1 \ge n_1 + n_2 - 1 = n$ proving the theorem.

In Figure 4.7 all considered combinations are illustrated. In this figure, there are only crossings of dotted and solid lines. Therefore, only these combinations can conflict with each other. By definition, the combination (Q_1, Q_2) forms a spanning tree of unique weight. On the other hand, no combination $(T^{(1)}_j, P_2 = T^{(2)}_\mu)$, $\alpha < j < \beta$ can conflict with a combination $(Q_1 = T^{(1)}_\alpha, T^{(2)}_\ell)$, $\mu < \ell < \nu$ since otherwise we can construct a partition spanning tree $T \neq Q$ with weight $w(Q)$ contradicting $\sigma(B, Q) = 1$. This proves the theorem. □

Hence, it is sufficient to consider only 2-vertex-connected bispanning graphs.

4.4 General Decomposition of Bispanning Graphs

As our study in the previous two sections shows, it is possible to combine partition spanning trees of bispanning (sub) graphs which are connected by a common cut vertex or which are connected by two edges. In this section, we want to generalize this study by giving a universal decomposition method for general (weighted) bispanning graphs. As a first step, given any weighted bispanning graph $B = (V, P, Q, w)$, we observe that it is possible to omit a pair (q, p) of multiple edges.

Proposition 4.7. *Let* $B = (V, P, Q, w)$ *be a weighted bispanning graph with* $\operatorname{ord}(B, P) < \operatorname{ord}(B, Q)$ *and* $\sigma(B, Q) = 1$*. Let* (p, q) *be a pair of multiple edges. Then, it holds that* $\operatorname{ord}'(B[q, p], Q[q, p]) < \operatorname{ord}'(B, Q)$*.*

Proof. The proof is analogously to the one of Theorem 4.3 depending on whether $w(q) < w(p)$ or conversely even if the edge connectivity of B is greater than two. □

Although Proposition 4.7 can be proven according to Theorem 4.3, there are also essential similarities to our analysis in Section 4.3. The key observation is that given a pair (q, p) of multiple edges connecting vertices v and w, the induced graph $B[V']$ on $V' = \{v, w\}$ is a bispanning graph itself. Following this idea, it is possible to prove Proposition 4.7 also by using the results of the last section.

Theorem 4.8. *Let* $B = (V, P, Q, w)$ *be a weighted bispanning graph such that* $\operatorname{ord}(B, P) < \operatorname{ord}(B, Q)$*,* $\sigma(B, Q) = 1$*. Let* B_1 *be a non-trivial bispanning subgraph of* B*. Let* B_2 *be the graph we obtain from* B *by contracting the subgraph* B_1 *to a single vertex. Then, Conjecture 7 holds for* B *if Conjecture 7 holds for* B_1 *and* B_2*.*

Proof. Given a weighted bispanning graph $B = (V, P, Q, w)$ with $\operatorname{ord}(B, P) < \operatorname{ord}(B, Q)$, $\sigma(B, Q) = 1$. Let $B_1 = (V', P_1, Q_1)$ be a non-trivial bispanning subgraph of B. Let $B_2 = (V'', P_2, Q_2)$ be the graph which we obtain from B by contracting B_1 to a single vertex, i.e., we have $B_2 = B/V_1$. We define two weight functions for B_1 and B_2 according to the function w. Hence, if $\sigma(B, Q) = 1$ holds then we have $\sigma(B_1, Q_1) = \sigma(B_2, Q_2) = 1$. Furthermore, we observe that B_2 indeed is a bispanning graph.

Remember that each partition spanning tree of B_1 together with each partition spanning tree of B_2 forms a partition spanning tree of the whole bispanning graph B. Thus, we are able to apply Lemma 4.5 or Lemma 4.6 depending on the weight difference between the remaining parts of Q and P with respect to B_1 and B_2. □

If a given bispanning graph is composite then there exists a non-trivial bispanning subgraph. In this case, we are able to apply Theorem 4.8. Suppose the bispanning graphs B_1 and B_2 are composite itself, we can recursively apply this theorem until the remaining bispanning graphs become atomic. Hence, we have to further analyze only atomic bispanning graphs in order to get a proof of Conjecture 7. The smallest non-trivial atomic bispannable graph is the complete graph on four vertices. This graph has (up to isomorphism) only one partition into two spanning trees which is analyzed in the following section.

4.5 Partitioning the K_4

In Section 4.2 and Section 4.3 we have seen that it is sufficient to analyse 2-vertex-connected and 3-edge-connected bispanning graphs. Moreover, by Theorem 4.8, it is possible to consider only simple graphs, i.e., graphs without multiple edges. The following proposition is due to Dirac [Dir52].

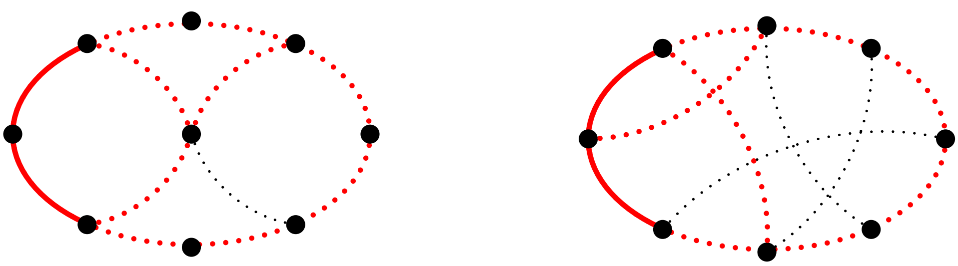
Proposition 4.9. *A 2-connected simple graph in which the degree of every vertex is at least* 3 *has a minor isomorphic to the complete graph* K_4.

Proof. Let $G = (V, E)$ be such a graph. A path P between two non-adjacent vertices v and w of a cycle C is called a *chord* if P is having only the vertices v and w in common with C. We claim that G contains a cycle of length at least 4 and each vertex of C is connected by a chord to another vertex in C.

Let $P = (v_1, v_2, \ldots, v_k)$ be a longest path in G. The vertex v_1 is adjacent to three vertices in P since otherwise there exists a longer path in G. Let v_i be the vertex adjacent to v_1 with largest index i. Then, the vertices $v_1, v_2, \ldots, v_i, v_1$ form a cycle of length at least 4. Now, we consider such a cycle $C = (v_1, v_2, \ldots, v_k, v_1)$ of length $k \geq 4$. Observe that v_1 is adjacent to another vertex w distinct to v_2 and v_k. If $w \in C$ then we are done. Hence, we assume $w \notin C$. Since v_1 is not a cut vertex, there exists a path P' between w and v_2 which does not pass v_1. Let v_i be the first vertex contained in C and lying on this path. If $i = 2$ or $i = k$, we have found a longer cycle contradicting the assumption that P is a longest path in G. Hence, the path between v_1 and v_i via w is a chord.

Now, we show the existence of a minor isomorphic to K_4 in G. To this end, we consider a cycle $C = (v_1, v_2, \ldots, v_k, v_1)$ of lenght $k \geq 4$ and distinguish two cases: either there are two chords connecting different pairs of vertices of C such that there is a common vertex not belonging to C or no two such chords of C have a vertex not belonging to C in common.

The first case is easy and is illustrated in Figure 4.8(a). For the second case, let v_i



(a) There are two chords with a common vertex. (b) No two chords have a common vertex.

Figure 4.8: Constructing a minor isomorphic to K_4.

and v_j be two vertices connected by a chord such that $j - i$ is minimal. Since v_i and v_j are two non-neighbouring vertices, there exists a vertex v_ℓ with $i < \ell < j$. Furthermore, the vertex v_ℓ is connected by a chord with a vertex lying on C between v_j and v_i. Then, the minor isomorphic to K_4 can be constructed according to Figure 4.8(b). $\square$

As a consequence, this proposition closes the gap to Section 3.3 because the cycle matroid of a bispanning graph is strongly base orderable if the graph has no minor isomorphic to the complete graph K_4. Hence, we have to focus on counting partition spanning trees in graphs which have a minor isomorphic to K_4. Moreover, the K_4 is the smallest bispanning graph with the property having no non-trivial subset $V' \subset V$ such that the subgraph induced on V' is a bispanning graph. Nevertheless, we will prove that Conjecture 7 holds even if the given weighted bispanning graph $B = (V, P, Q, w)$ is isomorphic to the complete graph on four vertices with a weight function that is required to satisfy $\text{ord}(B, P) < \text{ord}(B, Q)$ and $\sigma(B, Q) = 1$. Note that there is up to isomorphism only one assignment of the edges to two edge-disjoint spanning trees. We use the decomposition according to Figure 4.9 where the blue edges (a, b, and c) belong to Q and the red edges (d, e, and f) belong to P.

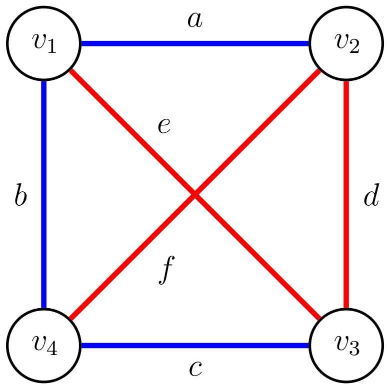


Figure 4.9: A decomposition of K_4 into two disjoint spanning trees.

Theorem 4.10. *Let $B = (V, P, Q)$ with $Q = \{a, b, c\}$ and $P = \{d, e, f\}$ be a partition of the K_4 into two disjoint spanning trees. Let $w\colon (P \cup Q) \to \mathbb{R}$ be a weight function such that* $\text{ord}(B, P) < \text{ord}(B, Q)$ *and* $\sigma(B, Q) = 1$. *Then, it holds that* $\text{ord}'(B, Q) \geq 4$.

Proof. Let $Q = \{a, b, c\}$ and $P = \{d, e, f\}$ according to Figure 4.9. It is easy to verify that this graph has 12 different partition spanning trees which are illustrated in Figure 4.10, where each pair of complementary trees is in a box with gray background. By the pigeon-hole principle, at least one partition spanning tree of each pair must have a weight smaller than $w(Q)$. We make a case distinction depending on the weights of the complementary spanning trees $\{a, c, d\}$ and $\{b, e, f\}$ in the upper right corner of Figure 4.10. These partition spanning trees do not have the property to be part of a path in the tree graph which corresponds to a so-called subsequence-interchangeable base ordering (cf. Figure 5.2 in Chapter 5). For the sake of readability, we associate with an edge q also its weight $w(q)$, i.e., $a+b+c$ is an abbreviation of $w(a)+w(b)+w(c)$.

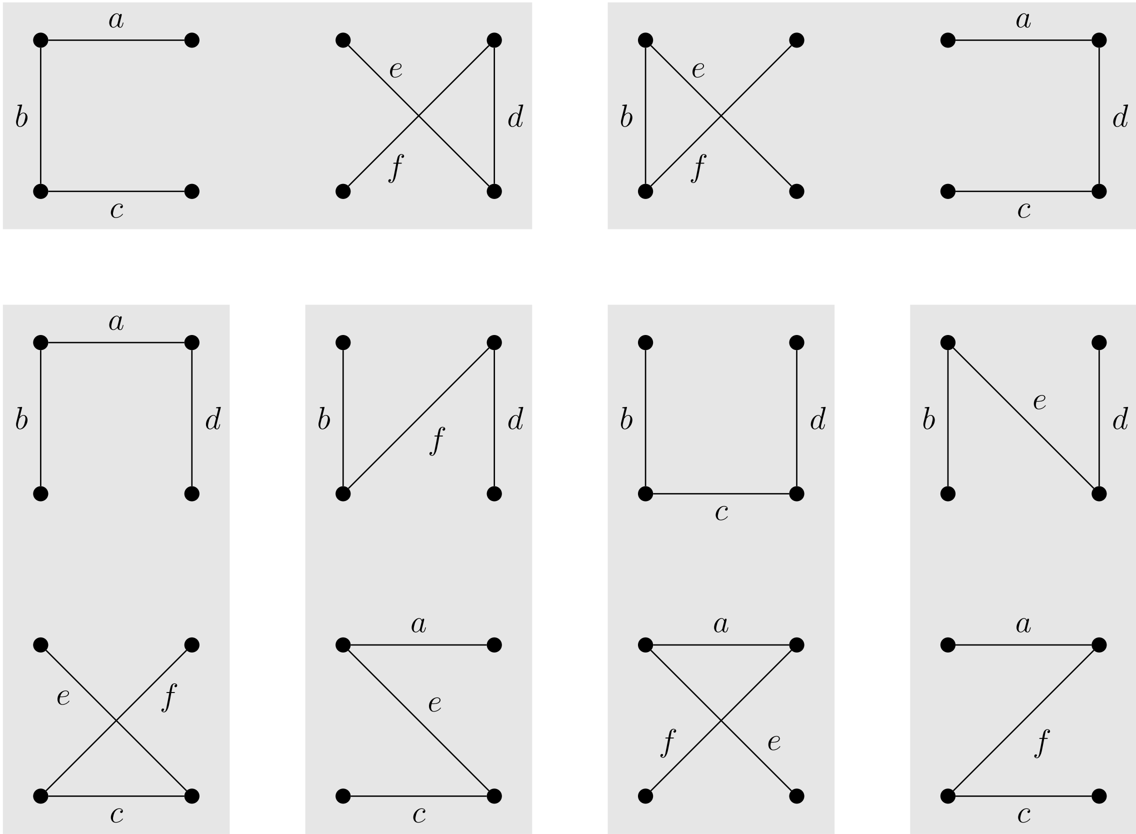


Figure 4.10: All partition spanning trees of the complete graph K_4.

First, we assume that $\{a, c, d\}$ and $\{b, e, f\}$ have the same weight which is smaller than $w(Q)$. Then, at least $\{a, e, f\}$ *and* $\{b, c, d\}$ or $\{a, c, f\}$ *and* $\{b, d, e\}$ must have different weights since otherwise we have

$$a + e + f = b + c + d \quad = \quad a + c + f = b + d + e \quad = \quad a + c + d = b + e + f$$

implying $w(c) = w(e)$. In this case, we obtain a contradiction to $\sigma(B, Q) = 1$ since $\{a, b, e\}$ will be a spanning tree with weight $w(Q) = a+b+c$. Hence, the spanning trees $\{a, c, d\}, \{d, e, f\}$, and at least one spanning tree in $\{\{a, e, f\}, \{b, c, d\}, \{a, c, f\}, \{b, d, e\}\}$ have distinct weights smaller than $w(Q)$ implying $\text{ord}'(B, Q) \geq 4$.

Now, assume the spanning trees $\{a, c, d\}$ and $\{b, e, f\}$ have different weights (where at least one of these weights is strictly smaller than $w(Q)$). If one of the remaining four pairs of partition spanning trees consists of trees with equal weights, we are done. Hence, we suppose that each pair consists of different weighted spanning trees implying the following two matrices of inequations:

$$\begin{pmatrix} a+b+d & \cdots & \neq & \cdots & b+d+f \\ \vdots & \ddots & & \iddots & \vdots \\ \neq & & ? & & \neq \\ \vdots & \iddots & & \ddots & \vdots \\ c+e+f & \cdots & \neq & \cdots & a+c+e \end{pmatrix}$$

and

$$\begin{pmatrix} b+c+d & \cdots & \neq & \cdots & b+d+e \\ \vdots & \ddots & & \iddots & \vdots \\ \neq & & ? & & \neq \\ \vdots & \iddots & & \ddots & \vdots \\ a+e+f & \cdots & \neq & \cdots & a+c+f \end{pmatrix} .$$

Here, the red inequality signs are *forced* since otherwise we obtain $w(a) = w(f)$ and $w(c) = w(e)$ contradicting $\sigma(B, Q) = 1$. The property that none of the four remaining pairs of partition spanning trees consists of trees with equal weights is marked by the green inequality signs. Furthermore, there are two question marks inside these matrices. If at least one of them can be replaced by '$\neq$' then we have three partition spanning trees with distinct weights less than $w(Q)$. Hence, we assume that both question marks have to be replaced by '$=$'. This means that both matrices contain at least one weight less than $w(Q)$. If these weights are different, we have $\text{ord}'(B, Q) \geq 4$. Otherwise, there are two different cases which have to be distinguished:

1. It holds that

$$c+e+f = b+d+f \quad = \quad b+c+d = a+c+f$$

   which is equivalent to

$$a+b+d = a+c+e \quad = \quad a+e+f = b+d+e$$

   since the sum of all edge weights is constant. Both of these equations imply $w(a) = w(e)$ and $w(c) = w(f)$, thus, the spanning tree $\{b, e, f\}$ has weight $w(Q)$ in contradiction to $\sigma(B, Q) = 1$.

2. It holds that

$$a+b+d = a+c+e \quad = \quad b+c+d = a+c+f$$

   which is equivalent to

$$c+e+f = b+d+f \quad = \quad a+e+f = b+d+e\,.$$

   If these equations do not conflict with either $a+c+d$ or $b+e+f$, we are done. Otherwise, we have to distinguish two further cases:

(a) We assume that

$$a + b + d = a + c + e = b + c + d = a + c + f \quad = \quad a + c + d$$

implying $a = b = c$ and $d = e = f$. Then, Q must be a maximum spanning tree and P a minimum spanning tree, respectively. In this case, the partition spanning trees $\{a, c, e\}$,$\{c, e, f\}$, and $\{d, e, f\}$ are sufficient to prove the claim.

(b) We assume that

$$a + b + d = a + c + e = b + c + d = a + c + f \quad = \quad b + e + f$$

implying $a = c$, $e = f$, $a + c = b + e$, and $c + d = e + f$. Clearly, we have $a = c \neq e = f$ since otherwise $\sigma(B, Q) > 1$. If $a = c < e = f$ holds then $a+c = b+e$ and $c+d = e+f$ imply $b < a = c$ and $d > e = f$ resulting in a contradiction to $w(Q) > w(P)$. Otherwise, we have $a = c > e = f$ together with $a + c = b + e$ and $c + d = e + f$ implying $b > a = c > e = f > d$, that is, Q is a maximum spanning tree and P a minimum spanning tree, respectively. Again, we obtain $\text{ord}'(B, Q) \geq 4$ and the claim follows. □

Since Conjecture 7 is at least as strong as Conjecture 6 (in fact we only consider a smaller class of spanning trees), we obtain the following corollary.

Corollary 4.11. *Let $B = (V, P, Q, w)$ be a weighted bispanning graph on four vertices such that $\text{ord}(B, P) < \text{ord}(B, Q)$ and $\sigma(B, Q) = 1$. Then, it holds that $\text{ord}(B, Q) \geq 4$.*

Chapter 5

Subsequence-Interchangeable Base Orderings

5.1 Introduction

Let $B = (V, P, Q)$ be a bispanning graph and let $m := |P| = |Q| = n - 1$ be the number of edges of a spanning tree in B, thus, the number of edges in B is $2m$. In general, the spanning trees P and Q are unordered edge sets. If P and Q are fixed then there are $m! \cdot m!$ different possibilities to label their edges by $p_1, \ldots, p_m$ and $q_1, \ldots, q_m$, respectively. We call a labeling of the edges of both spanning trees a *base ordering* and we denote the set of all such orderings by $\mathcal{BO}$. Given any ordering in $\mathcal{BO}$, we usually write it by concatenating the edges of Q and P

$$q_1\, q_2\, \cdots\, q_m\, p_1\, p_2\, \cdots\, p_m \tag{5.1}$$

or using the ordered pairs

$$(q_1, p_1), (q_2, p_2), \ldots, (q_m, p_m)\ . \tag{5.2}$$

The latter representation is useful because we can interpret the pairs as edge swaps. Hence, we call (5.2) the *edge swap sequence* of the corresponding base ordering whereas (5.1) is called *base ordering notation*. Note that a single edge swap of a base ordering does not have to be an edge swap as defined in Chapter 2 since the sets might not form spanning trees after exchanging the edges. In the context of matroid theory, a variant of base orderings has been studied by Kajitani, Ueno, and Miyano [KUM88]. Speaking in terms of spanning trees in a graph $G = (V, E)$, they proved that the edges in E can be (not necessarily cyclically) ordered in such a way that any $n - 1$ consecutive edges of this ordering form a spanning tree of G.

In the following, we establish a connection between base orderings and tree graphs. To this end, we remember the definition of the tree graph $\mathcal{G}(G) = (\mathcal{V}, \mathcal{E})$ of an arbitrary connected graph $G = (V, E)$: the vertex set $\mathcal{V} = \mathcal{T}(G)$ consists of all spanning trees in G and two spanning trees $T, T' \in \mathcal{T}(G)$ are connected by an edge $\{T, T'\} \in \mathcal{E}$ if and only if they are related by an edge swap. Given two arbitrary spanning trees T and T', there exists a (not necessarily unique) path $\mathcal{P}$ between them (implied by Lemma 3.1). For the sake of readability, we denote a path (in a tree graph) by the corresponding sequence of edges (instead of using the vertices). Each of these edges represents some edge swap $s = (e, f)$. If T and T' differ in exactly k edges (i.e., the symmetric difference $T \Delta T' = (T \setminus T') \cup (T' \setminus T)$ contains $2k$ edges), there exists a path $\mathcal{P} = (s_1, \ldots, s_k)$ of length k joining T and T' in $\mathcal{G}(G)$. We define the operation $\mathcal{P}(T) = (s_1 \circ \ldots \circ s_k)(T) = T'$ to be the *application of* $\mathcal{P}$ *to* T. Moreover, we define $\mathcal{P}^{-1} = (s_1^{-1}, \ldots, s_k^{-1})$ where $s^{-1} = (f, e)$ if $s = (e, f)$.

Applying edge swap sequences to spanning trees can be generalized as follows. Let T be a spanning tree of G and let $S = (s_1, \ldots, s_k)$ be a sequence of edge swaps such that $s_i = (e_i, f_i)$, $e_i \in T$, and $f_i \notin T$ for all $1 \leq i \leq k$. Then, we denote by

$$S(T) = (s_1 \circ \ldots \circ s_k)(T) = (T \setminus \{e_1, \ldots, e_k\}) \cup \{f_1, \ldots, f_k\}$$

the set which is obtained by applying each edge swap to T. Note that the set $S(T)$ does not have to form a spanning tree. If $S(T)$ yields a spanning tree, we call S an *admissible sequence* with respect to T.

Given a bispanning graph $B = (V, P, Q)$, we denote by $\mathcal{PQ}$ the set of all paths in the tree graph $\mathcal{G}(B)$ of length m between P and Q. By our above analysis, each path corresponds to an ordering but an ordering does not necessarily correspond to some path. Hence, we obtain the relationship

$$\mathcal{PQ} \subseteq \mathcal{BO} .$$

In the following sections of this chapter, we will further classify the set $\mathcal{PQ}$ of all paths between the edge-disjoint spanning trees P and Q. More precisely, we present two properties which we call *cyclic* and *subsequence-interchangeable* (cyclic base orderings were already studied in [Wie06]). Given the set $\mathcal{CBO}$ of all cyclic base orderings of a bispanning graph $B = (V, P, Q)$ and given the set $\mathcal{SIBO}$ of all subsequence-interchangeable base orderings of B, we show the following relationship between these sets:

$$\mathcal{SIBO} \subseteq \mathcal{CBO} \subseteq \mathcal{PQ} \subseteq \mathcal{BO} .$$

At the end of this chapter, we show a connection between subsequence-interchangeable base orderings and the findings of Chapters 3 and 4.

5.2 Cyclic Base Orderings

Definition 5.1. *Let $B = (V, P, Q)$ be a bispanning graph. An ordering of the edges of $P = \{p_1, \ldots, p_m\}$ and $Q = \{q_1, \ldots, q_m\}$ is called a cyclic base ordering (or short CBO) if any m cyclically consecutive elements in*

$$q_1\ q_2\ \ldots\ q_m\ p_1\ p_2\ \ldots\ p_m \tag{5.3}$$

form a spanning tree of B.

As aforementioned, any base ordering can be represented by a sequence $S = (s_1, \ldots, s_m)$ of edge swaps with $s_i = (q_i, p_i)$ for all $i = 1, \ldots, m$. Using this representation, an edge swap sequence corresponds to a cyclic base ordering if and only if the sets

$$T_Q = (s_1 \circ \ldots \circ s_i)(Q) \quad \text{and} \quad T_P = (s_1^{-1} \circ \ldots \circ s_i^{-1})(P)$$

are spanning trees of B for all $i = 1, \ldots, m$.

The existence of CBOs for all partitions of a bispannable graph into two spanning trees was already proven in [Wie06]. Nevertheless, we take a more detailed look at the proof in order to use several ideas of it when studying subsequence-interchangeable base orderings in the following section.

Theorem 5.2. *There is a cyclic base ordering for each bispanning graph $B = (V, P, Q)$.*

Proof. We prove this theorem by induction over the size m of a spanning tree in B. If $m = 1$ the bispanning graph B consists of two vertices connected by two parallel edges, $p \in P$ and $q \in Q$. Clearly, each element of the sequence

$$q \; p$$

forms a spanning tree of B, thus, we have found a cyclic base ordering of B. Now, let $m > 1$ and $B^{(m)} = (V, P^{(m)}, Q^{(m)})$ be a bispanning graph on $m + 1$ vertices. By Lemma 2.2, $B^{(m)}$ contains a vertex v of degree $\deg(v) = 2$ or $\deg(v) = 3$.

First, let v be a vertex incident to exactly two edges $q_m \in Q^{(m)}$ and $p_m \in P^{(m)}$. We observe that these edges are leaves of $Q^{(m)}$ and $P^{(m)}$, respectively. Hence, by cutting off these two edges and removing the vertex v (we reverse a double-leaf attachment), the remaining graph is a bispanning graph again. Let $B^{(m-1)}$ be this graph consisting of the edge-disjoint spanning trees $P^{(m-1)}$ and $Q^{(m-1)}$ with $P^{(m-1)} = P^{(m)} \setminus \{p_m\}$ and $Q^{(m-1)} = Q^{(m)} \setminus \{q_m\}$.

By induction hypothesis, the bispanning graph $B^{(m-1)}$ has a CBO, say

$$q_1 \; q_2 \; \cdots \; q_{m-1} \; p_1 \; p_2 \; \cdots \; p_{m-1} \; .$$

Now, we put the edge q_m at any position into the sequence of $Q^{(m-1)}$ and p_m at the corresponding position of $P^{(m-1)}$. For example, the sequence

$$q_1 \; q_2 \; \cdots \; q_{m-1} \; q_m \; p_1 \; p_2 \; \cdots \; p_{m-1} \; p_m \tag{5.4}$$

is a CBO since each cyclically consecutive subsequence of length m of (5.4) contains either q_m or p_m but not both of them. Hence, we append only a leaf to all considered spanning trees of $B^{(m-1)}$ maintaining the property of being a spanning tree.

If $B^{(m)}$ does not contain any vertex of degree two, there exists a vertex v of degree $\deg(v) = 3$. We distinguish two cases depending on whether v is incident to two edges of $P^{(m)}$ and one edge of $Q^{(m)}$, or v is incident two edges of $Q^{(m)}$ and one edge of $P^{(m)}$.

For the first case, let v be incident to p_{m-1}, p_m, and q_m. We construct a new bispanning graph $B^{(m-1)}$ by reversing the edge-split with respect to q_m: we remove the vertex v and its adjacent edges and we introduce a new edge $\tilde{p}$ connecting the vertices of p_{m-1} and p_m opposite to v (see Figure 5.1 for an illustration). Observe that this operation equals a contraction of either p_{m-1} or p_m and subsequently discarding the edge q_m. Clearly, the remaining edges $Q^{(m-1)} = Q^{(m)} \setminus \{q_m\}$ form a spanning

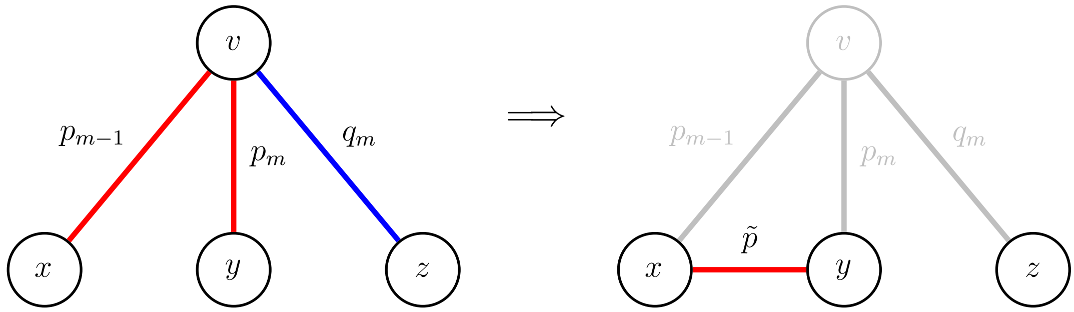


Figure 5.1: Reversing an edge-split of P.

tree since we cut off a leaf of $Q^{(m)}$ by removing q_m. Furthermore, the set $P^{(m-1)} = (P^{(m)} \setminus \{p_{m-1}, p_m\}) \cup \{\tilde{p}\}$ is a spanning tree, too. This follows since the contraction of any tree-edge (an edge belonging to the considered tree) maintains the property of being a spanning tree.

By induction hypothesis, there exists a cyclic base ordering for $B^{(m-1)}$, say

$$q_1 \ \ldots \ q_{i-1} \, \tilde{q} \, q_i \ \ldots \ q_{m-2} \, p_1 \ \ldots \ p_{i-1} \, \tilde{p} \, p_i \ \ldots \ p_{m-2} \, . \tag{5.5}$$

Let $\tilde{q}$ be the corresponding edge to $\tilde{p}$. Now, we insert q_m *immediately after* $\tilde{q}$ and claim that replacing $\tilde{p}$ either by p_{m-1} p_m or by p_m p_{m-1} produces a CBO for $B^{(m)}$. The corresponding sequence is

$$q_1 \ \ldots \ q_{i-1} \, \tilde{q} \, q_m \, q_i \ \ldots \ q_{m-2} \, p_1 \ \ldots \ p_{i-1} \left[p_{m-1} \, p_m \right] p_i \ \ldots \ p_{m-2} \tag{5.6}$$

where the brackets mean that the order of p_{m-1} and p_m is not fixed yet. In the following, we prove that there exists an order of p_{m-1} and p_m such that (5.6) is a CBO (more precisely, exactly one of the two possibilities yields a CBO). Thus, we have to consider any m cyclically consecutive elements of (5.6) and verify that they indeed form a spanning tree.

First, we observe that any m cyclically consecutive elements of (5.6) which do not include p_{m-1} and p_m (but the edge q_m) form a spanning tree because the corresponding subsequence in (5.5) (without q_m) forms a spanning tree and q_m only connects the leaf-vertex v. If the sequence of m cyclically consecutive elements of (5.6) contains both edges, p_{m-1} and p_m, the corresponding subsequence in (5.5) contains the edge $\tilde{p}$, thus, we only *stretched* the edge $\tilde{p}$ obtaining again a spanning tree. It remains to show that either

$$\{q_m, q_i, \ldots, q_{m-2}, p_1, \ldots, p_{m-1}\} \tag{5.7}$$

or

$$\{q_m, q_i, \ldots, q_{m-2}, p_1, \ldots, p_m\} \tag{5.8}$$

is a spanning tree. We observe that the complementary sets of (5.7) and (5.8) are spanning trees since the only difference to the spanning tree $\{q_1, \ldots, q_{i-1}, \tilde{q}, q_m, p_i, \ldots, p_{m-2}\}$ is the edge joining the leaf v.

In order to prove that either (5.7) or (5.8) is a spanning tree, we observe that by assumption, the set

$$T = \{p_{m-1}, p_m, p_i, \ldots, p_{m-2}, q_1, \ldots, q_{i-1}\}$$

is a spanning tree of $B^{(m-1)}$. Now, we consider the fundamental cycle $C(T, q_m)$ of $B^{(m)}$ defined by q_m with respect to T. This cycle *must* contain the edge q_m and therefore pass the vertex v. Hence, either p_{m-1} or p_m (*exactly one*) is part of the cycle. Hence, the leaving edge of the corresponding edge swap is clearly defined. This proves that either (5.7) or (5.8) forms a spanning tree.

We remark that it is also possible to insert q_m *immediately before* $\tilde{q}$ in (5.5). The resulting ordering is

$$q_1 \ \ldots \ q_{i-1} \ q_m \ \tilde{q} \ q_i \ \ldots \ q_{m-2} \ p_1 \ \ldots \ p_{i-1} \left[p_{m-1} \ p_m \right] p_i \ \ldots \ p_{m-2} \ .$$

In this case, the order of p_{m-1} and p_m is determined depending on whether

$$\{p_{m-1}, p_i, \ldots, p_{m-2}, q_1, \ldots, q_m\}$$

or

$$\{p_m, p_i, \ldots, p_{m-2}, q_1, \ldots, q_m\}$$

is a spanning tree. Again, not both sets can be spanning trees.

To conclude the proof, we assume that v is incident to two edges of $Q^{(m)}$ and one edge of $P^{(m)}$, say q_{m-1}, q_m, and p_m. Analogous to the previous case, we construct a bispanning graph $B^{(m-1)}$ by reversing the edge-split with respect to p_m. This situation can be illustrated analogously to Figure 5.1. By induction hypothesis, there is a cyclic base ordering for $B^{(m-1)}$, say

$$q_1 \ \ldots \ q_{i-1} \ \tilde{q} \ q_i \ \ldots \ q_{m-2} \ p_1 \ \ldots \ p_{i-1} \ \tilde{p} \ p_i \ \ldots \ p_{m-2} \tag{5.9}$$

which can be extended analogously to the previous case to two different CBOs: Insert the edge p_m *exactly before* or *exactly after* $\tilde{p}$. Again, the order of q_{m-1} and q_m is clearly defined and follows from the structure (i.e., the underlying cycles) of the bispanning graph. $\square$

Clearly, each cyclic base ordering of a bispanning graph $B = (V, P, Q)$ corresponds to some path between P and Q in the tree graph $\mathcal{G}(B)$. Hence, we obtain

$$\mathcal{CBO} \subseteq \mathcal{PQ} \ .$$

5.3 Subsequence-Interchangeable Base Orderings

As mentioned above, there is a refined variant of cyclic base orderings. This new kind of base ordering is formally defined as follows.

Definition 5.3. *Let $B = (V, P, Q)$ be a bispanning graph. We call a sequence of edge swaps $S = (s_1, s_2, \dots, s_m)$ transforming Q into P a* subsequence-interchangeable base ordering *(or short SIBO) if $S' = (s_i, \dots, s_j)$ is an admissible sequence with respect to Q for all $1 \leq i \leq j \leq m$, i.e., applying any S' to Q yields a spanning tree.*

Hence, an admissible sequence of edge swaps S transforming Q into P is a subsequence-interchangeable base ordering if each (consecutive) subsequence of S is admissible *with respect to* Q, too. Note that these subsequences do not have to be admissible with respect to P (an example is given on the next page). First, we prove that each SIBO also is a CBO implying

$$\mathcal{SIBO} \subseteq \mathcal{CBO} \, .$$

Proposition 5.4. *Let $B = (V, P, Q)$ be a bispanning graph and let S be a subsequence-interchangeable base ordering for B. Then, S is a cyclic base ordering for B.*

Proof. Let $B = (V, P, Q)$ be a bispanning graph and let $S = (s_1, \dots, s_m)$ be a subsequence-interchangeable base ordering of B with $s_k = (q_k, p_k)$ for all $1 \leq k \leq m$. By our remark after Definition 5.1, we have to prove that the sets $T_Q = (s_1 \circ \dots \circ s_k)(Q)$ and $T_P = (s_1^{-1} \circ \dots \circ s_k^{-1})(P)$ are spanning trees of B for all $1 \leq k \leq m$.

The first part follows directly by Definition 5.3 choosing $i = 1$ and $j = k$. For the second part, we choose $i = k + 1$ and $j = m$. Then, we obtain

$$\begin{aligned} T_P &= (s_1^{-1} \circ \dots \circ s_k^{-1})(P) \\ &= (P \setminus \{p_1, \dots, p_k\}) \cup \{q_1, \dots, q_k\} \\ &= (Q \setminus \{q_{k+1}, \dots, q_m\}) \cup \{p_{k+1}, \dots, p_m\} \\ &= (s_{k+1} \circ \dots \circ s_m)(Q) \end{aligned}$$

which is a spanning tree, too. This proves the claim. □

Reversing the sequence of edge swaps of each SIBO preserves the property of being subsequence-interchangeable. Note that cyclic base orderings have this property, too.

Proposition 5.5. *Let $B = (V, P, Q)$ be a bispanning graph. If a sequence of edge swaps $S = (s_1, s_2, \dots, s_m)$ is a subsequence-interchangeable base ordering then the reverse sequence $S_r = (s_m, s_{m-1}, \dots, s_2, s_1)$ is a SIBO, too.*

Proof. Let S be a subsequence-interchangeable base ordering of B. Then, any subsequence of S_r:

$$s_m, \dots, s_{i+1}, s_i, s_{i-1}, \dots, s_{j+1}, s_j, s_{j-1}, \dots, s_1$$

also appears in S (in reverse direction but the sets are the same):

$$s_1, \dots, s_{j-1}, s_j, s_{j+1}, \dots, s_{i-1}, s_i, s_{i+1}, \dots, s_m \, .$$ □

Since the complete graph on four vertices K_4 plays a central role in the previous two chapters, we will consider its cyclic base orderings in order to find an ordering which satisfies the stronger property. According to the partition given in Figure 4.9 on Page 55, the K_4 has the following 8 cyclic base orderings

$$\begin{align}
&(a,d),\ (c,f),\ (b,e) \nonumber\\
&(b,e),\ (c,f),\ (a,d) \nonumber\\
&(c,d),\ (a,e),\ (b,f) \nonumber\\
&(b,f),\ (a,e),\ (c,d) \nonumber\\
&(a,d),\ (c,e),\ (b,f) \tag{5.10}\\
&(b,f),\ (c,e),\ (a,d) \tag{5.11}\\
&(c,d),\ (a,f),\ (b,e) \tag{5.12}\\
&(b,e),\ (a,f),\ (c,d)\ . \tag{5.13}
\end{align}$$

Actually, the latter 4 CBOs (5.10) through (5.13) even have the property of being subsequence-interchangeable. Hence, the set $\mathcal{CBO}$ of all cyclic base orderings of a bispanning graph is (in general) a proper superset of $\mathcal{SIBO}$. The paths of the K_4's tree graph corresponding to subsequence-interchangeable base orderings are illustrated in more detail in Figure 5.2. In Figure 5.3, we can see the tree graph of the K_4 restricted to partition spanning trees (note that this subgraph is always connected [FRS85]). In this figure, the paths corresponding to SIBOs are colored in red whereas the blue paths are only cyclic base orderings of the K_4 (the red/blue dashed edges belong to red and blue paths).

In Definition 5.3, we say that each subsequence of a SIBO S applied to Q yields a spanning tree. We observe that these subsequences do not need to be admissible with respect to P. For example, take a look at the four SIBOs of the K_4: if we apply the middle edge swaps (c,e) or (a,f) (actually the inverse of them) to P then we do not obtain a spanning tree.

A question which arises now is whether or not each bispanning graph has a SIBO. Although we do not know the correct answer to this question, we conjecture that each bispanning graph has a SIBO since there are strong indications that this is true. In the remainder of this chapter, we study several possiblities in order to make progress in finding a proof for the following conjecture.

Conjecture 8. *There exists a subsequence-interchangeable base ordering for each bispanning graph $B = (V, P, Q)$.*

We pursue two different approaches. On the one hand, we consider the so-called bottom-up approach. Here, we analyze the two different operations, 'edge-split' and 'double-leaf attachment' (see Chapter 2), which can be used to create a bispanning graph as shown in Theorem 2.6. We show that under certain assumptions, it is possible to construct a subsequence-interchangeable base ordering for a graph which is obtained by a sequence of these operations starting with a single vertex. On the other hand,

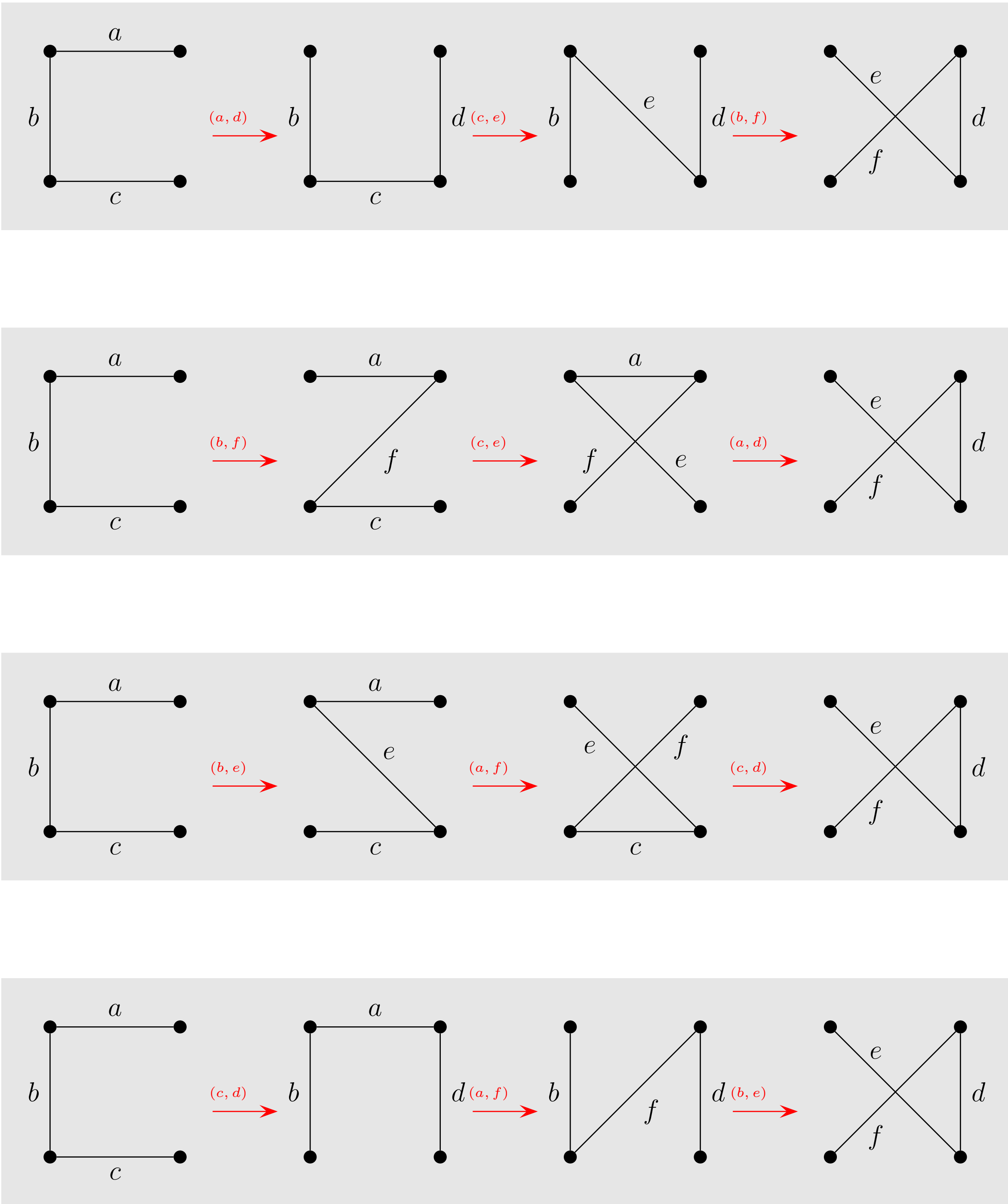


Figure 5.2: All paths of the K_4's tree graph correspoding to SIBOs.

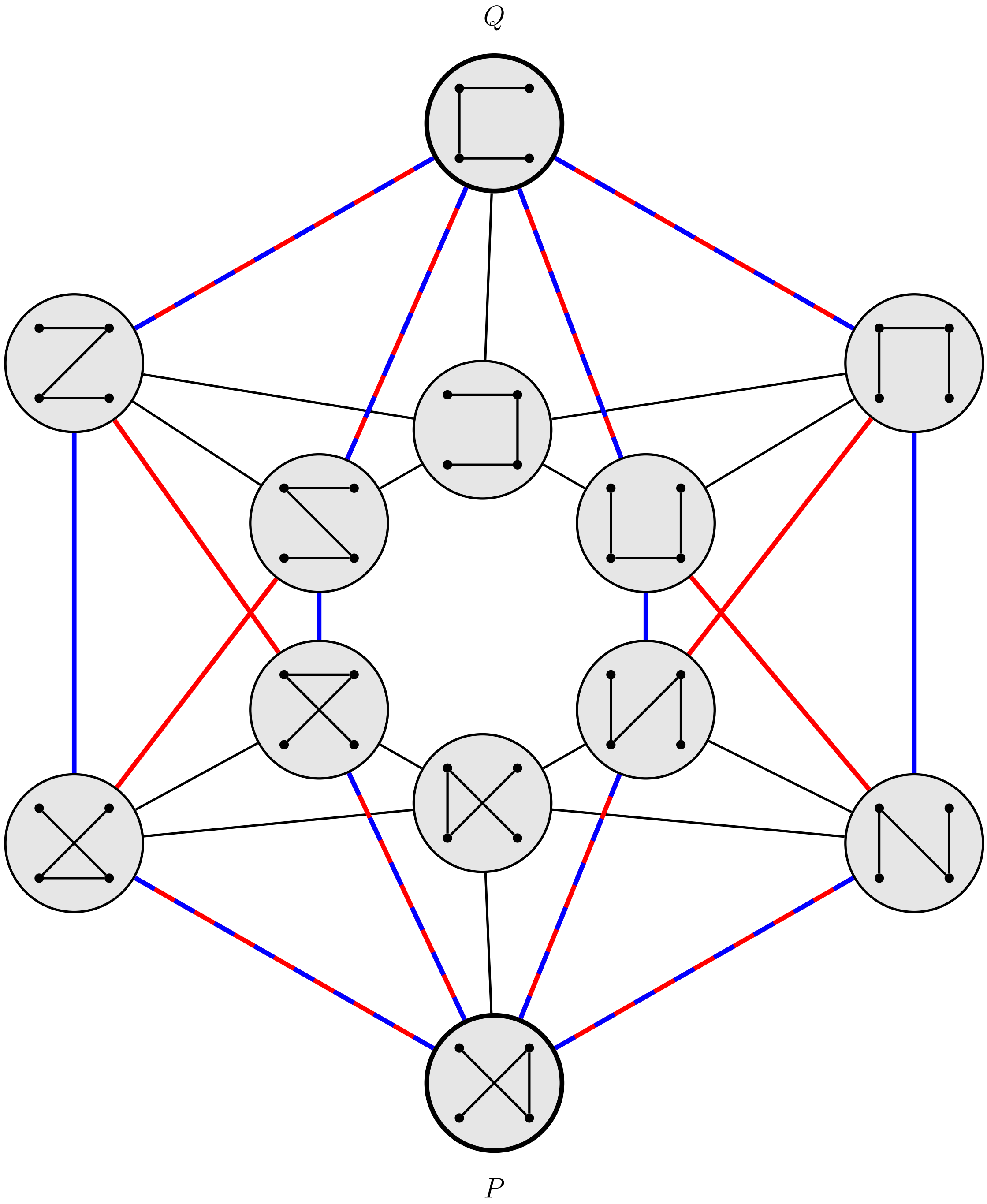


Figure 5.3: The K_4's tree graph restricted to partition spanning trees. The red paths are SIBOs whereas the blue paths are only CBOs of the K_4 (with respect to the partition into Q and P).

we consider a top-down approach: we analyze a given bispanning graph in order to find a suitable decomposition for constructing a SIBO. The latter approach uses some ideas of Chapter 4. Subsequently, we prove that each partition of a wheel graph into two spanning trees has a subsequence-interchangeable base ordering. Moreover, we can compute such a SIBO in linear time.

5.3.1 Bottom-Up Approach

In Chapter 2 (see Theorem 2.6), we have seen that each bispanning graph can be constructed using a sequence of 'double-leaf attachment' and 'edge-split' operations. Now, we analyze their impact on subsequence-interchangeable base orderings.

Theorem 5.6. *Let $B = (V, P, Q)$ be a bispanning graph and $S = ((q_1, p_1), \dots, (q_m, p_m))$ be a subsequence-interchangeable base ordering of B (with respect to Q). Then, there exists a SIBO for the bispanning graph $B' = (V', P', Q')$ which is obtained by applying a 'double-leaf attachment' to any two vertices u and v in V.*

Proof. The 'double-leaf attachment' operation introduces a new vertex w and connects it with two edges to B. We denote these new edges by q_{m+1} and p_{m+1} depending on the spanning tree to which they belong. Then

$$S' = ((q_1, p_1), \dots, (q_m, p_m), (q_{m+1}, p_{m+1}))$$

is a subsequence-interchangeable base ordering for the resulting bispanning graph B'. This follows easily by the observation that both edges connect the leaf w with respect to the spanning trees P' and Q'. Since these new edges are exchanged by each other, they do not appear together in any considered edge set. Hence, each subsequence of S' is admissible with respect to Q'. □

We remark that it is possible to insert the edge swap (q_{m+1}, p_{m+1}) at any position in the sequence S in order to obtain a new subsequence-interchangeable base ordering of B'. Hence, the number of SIBOs of B' is by a factor of $m+1$ larger than the number of SIBOs of B if a double-leaf attachment operation is applied to any two vertices of B.

The edge-split operation is not as easy as a double-leaf attachment. Here, we restrict ourselves to splitting only special edges. The reason for this is not obvious and will be explained in more detail later.

Theorem 5.7. *Let $B = (V, P, Q)$ be a bispanning graph and $S = ((q_1, p_1), \dots, (q_m, p_m))$ be a subsequence-interchangeable base ordering of B (with respect to Q). Then, there exists a SIBO for the bispanning graph $B' = (V', P', Q')$ which is obtained by splitting one of the edges in $\{q_1, p_1, q_m, p_m\}$ by any vertex in V.*

Proof. Because of Proposition 5.5, i.e., the symmetry of SIBOs, it suffices to consider only the case of splitting q_1 and p_1. We start the proof by analyzing the latter splitting operation, that is, the edge p_1 is 'split by an edge q_0' into two edges $p_{1,0}$ and $p_{1,1}$ where the labeling of the new edges is determined as follows. Exactly one of these new edges is contained in the fundamental cycle (of B')

$$C((Q' \setminus \{q_0, q_1\}) \cup \{p_{1,0}, p_{1,1}\}, q_0) \; .$$

This edge is labeled by $p_{1,0}$ and the other part is labeled by $p_{1,1}$ (see Figure 5.4).

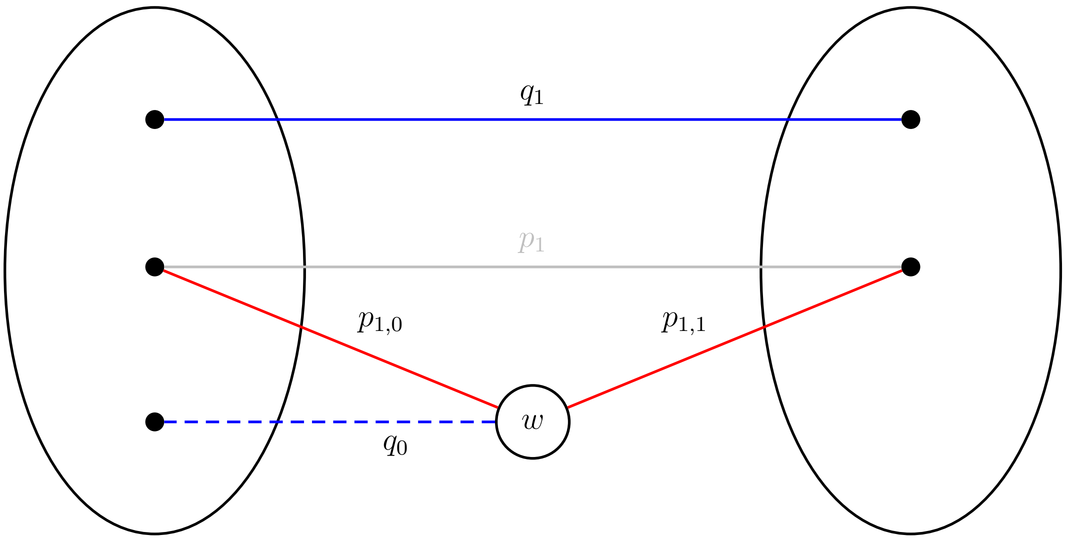


Figure 5.4: Splitting an edge of P. Connected components of B' with respect to $Q' \setminus \{q_1\}$.

Now, we claim that

$$(q_1, p_{1,1}), (q_0, p_{1,0}), (q_2, p_2), \dots, (q_m, p_m) \tag{5.14}$$

is a subsequence-interchangeable base ordering for the resulting bispanning graph B'. First, we observe that each subsequence $(q_i, p_i), \dots, (q_j, p_j)$ of (5.14) with $2 \le i \le j \le m$ is an admissible sequence with respect to Q'. This follows easily by the assumption that S is a SIBO for B. More precisely, the considered spanning trees differ only in the edge q_0 which connects the leaf w (with respect to Q'). Furthermore, the whole sequence (5.14) is admissible. It remains to show that each sequence starting with $(q_0, p_{1,0})$ is admissible. Applying the single edge swap $(q_0, p_{1,0})$ to Q' clearly yields a spanning tree. Suppose there exists an index $j \in \{2, \dots, m\}$ such that the subsequence $(q_0, p_{1,0}), (q_2, p_2), \dots, (q_j, p_j)$ is not admissible with respect to Q'. By assumption the same subsequence without the first edge swap $(q_0, p_{1,0})$ is admissible. Let Q'_j be the corresponding spanning tree, i.e.,

$$Q'_j = (Q' \setminus \{q_2, \dots, q_j\}) \cup \{p_2, \dots, p_j\} \; . \tag{5.15}$$

We observe that w is a leaf with respect to Q'_j and q_0 is the corresponding edge. Then, exchanging q_0 and $p_{1,0}$ obviously yields a spanning tree in contradiction to the assumption.

Now, we consider a split operation of q_1 into $q_{1,0}$ and $q_{1,1}$ by an edge p_0. Let $q_{1,0}$ be the edge belonging to the fundamental cycle $C(Q', p_0)$ of B'. Then, we claim that

$$(q_{1,0}, p_0), (q_{1,1}, p_1), (q_2, p_2), \ldots, (q_m, p_m) \tag{5.16}$$

is a SIBO of B'. As already seen in the previous case, most subsequences of (5.16) are admissible with respect to Q' (more precisely, each subsequence starting with an edge swap (q_i, p_i), $2 \leq i \leq m$, and the whole sequence are admissible with respect to Q'). Again, the crucial point is to show that any subsequence starting with the second edge swap $(q_{1,1}, p_1)$ is admissible with respect to Q'. Analogous to the previous case, we assume that there exists an index j such that the subsequence $(q_{1,1}, p_1), (q_2, p_2), \ldots, (q_j, p_j)$ is not admissible with respect to Q'. Thus, we assume that

$$(Q' \setminus \{q_{1,1}, q_2, \ldots, q_j\}) \cup \{p_1, p_2, \ldots, p_j\} \tag{5.17}$$

contains a cycle. We observe that without exchanging the edges of the pair $(q_{1,1}, p_1)$, the

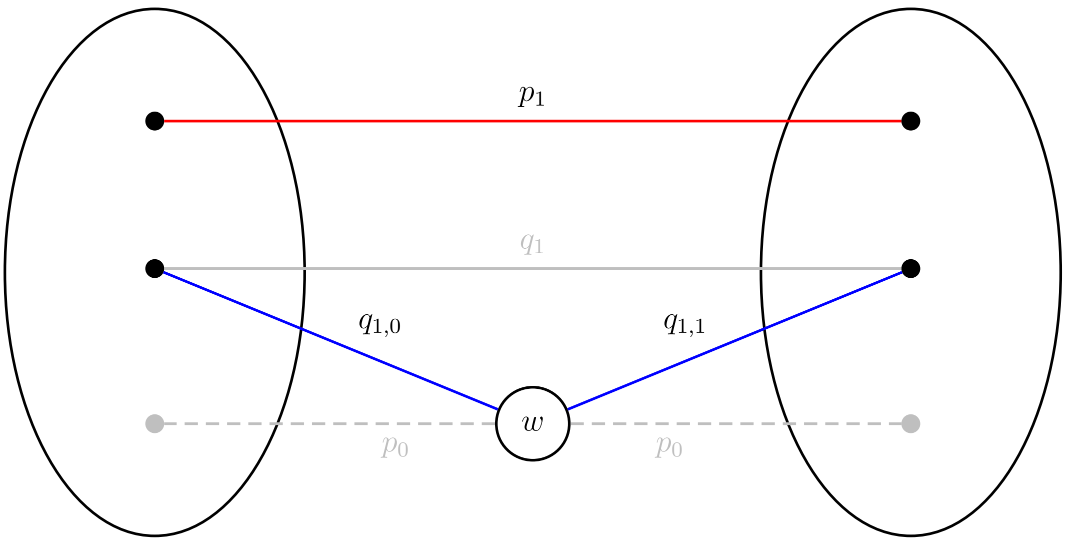


Figure 5.5: Splitting an edge of Q. The edge p_1 crosses the cut formed by q_1 with respect to Q'_j.

resulting set yields a spanning tree. Let Q'_j denote this spanning tree (cf. (5.15)). By assumption, (q_1, p_1) is an admissible edge swap with respect to $Q_j = (Q \setminus \{q_2, \ldots, q_j\}) \cup \{p_2, \ldots, p_j\}$ (in the bispanning graph B). Hence, the edge p_1 crosses the cut formed by removing q_1 with respect to Q_j. Then, both edge swaps $(q_{1,0}, p_1)$ and $(q_{1,1}, p_1)$ will be admissible with respect to Q'_j in contradiction to (5.17) containing a cycle. This situation is illustrated in Figure 5.5 where the splitting edge p_0 can either connect w with the left or the right component. $\square$

Splitting edges in the middle of a SIBO

If we look at the proof of Theorem 5.7, it is not obvious why there exists the restriction of splitting only a 'first' (i.e., q_1 or p_1) or a 'last' (i.e., q_m or p_m) edge of a given subsequence-interchangeable base ordering. Here, we note that these edges are sometimes called 'boundary' edges. To understand this restriction, we have to take a closer look at our ideas which are related to the proof of Theorem 5.2. Actually, the original idea to construct a SIBO was to follow the same lines as constructing a CBO, i.e., using a proof by induction. Clearly, the induction hypothesis holds also for the stronger property (i.e., SIBO instead of CBO). Furthermore, if the considered bispanning graph contains a vertex of degree 2, we can easily extend a given SIBO S (cf. Theorem 5.6). In particular, it is possible to construct a lot of different SIBOs when applying a double-leaf attachment since the corresponding edge swap can be integrated at any position in S. The crucial operation is the edge-split. Here, we only have two different possibilities according to the proof of Theorem 5.2. We can insert the splitting edge either *immediatly after* or *immediately before* an edge $\tilde{q}$ (the corresponding swap edge of the splitted edge). In both cases, the two parts of the splitted edge can be ordered such that the resulting sequence yields a CBO. If we assume that the given ordering is even subsequence-interchangeable then exactly one of these extensions yields a SIBO provided that we split a special edge. This case was used in Theorem 5.7 were a special edge is a 'boundary' edge of the given SIBO. Splitting a 'middle' edge (i.e., an edge q_i or p_i with $1 < i < m$) of a SIBO does not always yield a SIBO for the resulting bispanning graph. We further analyze this situation in the following and give some examples of bispanning graphs such that it is not possible to split a 'middle' edge although the presented graphs have a lot of different SIBOs.

Let $B^{(m)} = (V, P, Q)$ be a bispanning graph on $m + 1$ vertices. Suppose that $B^{(m)}$ does not contain a vertex of degree 2. Then, there exists a vertex v of degree $\deg(v) = 3$ by Lemma 2.2. Without loss of generality, we assume that v is incident to one edge $q_m \in Q$ and two edges $p_{m-1}, p_m \in P$. Let $B^{(m-1)}$ be the bispanning graph obtained by reversing the edge-split with respect to q_m (cf. Figure 5.1). Suppose that this graph has a SIBO

$$(q_1, p_1), \ldots, (q_{i-1}, p_{i-1}), (\tilde{q}, \tilde{p}), (q_i, p_i), \ldots, (q_{m-2}, p_{m-2}) \tag{5.18}$$

or using the base ordering notation

$$q_1 \ \ldots \ q_{i-1} \ \tilde{q} \ q_i \ \ldots \ q_{m-2} \ p_1 \ \ldots \ p_{i-1} \ \tilde{p} \ p_i \ \ldots \ p_{m-2} \,.$$

Let $\tilde{p}$ be the edge which has to be split in order to obtain $B^{(m)}$ and let $\tilde{q}$ the corresponding edge of its edge swap. Since each SIBO has to be a CBO, we use the idea mentioned above and introduce the splitting edge q_m exactly before and/or exactly after the edge $\tilde{q}$ in order to obtain a CBO. The order of p_{m-1} and p_m depends on the structure of the bispanning graph. The proof of Theorem 5.2 contains more details about finding the right order of these edges. Hence, we suppose that both edges are labeled such that

$$q_1 \, q_2 \ \ldots \ q_{i-1} \, q_m \, \tilde{q} \, q_i \ \ldots \ q_{m-2} \, p_1 \, p_2 \ \ldots \ p_{i-1} \, p_{m-1} \, p_m \, p_i \ \ldots \ p_{m-2} \,. \tag{5.19}$$

is a cyclic base ordering. Now, there are two cases. Either

$$q_1\ q_2\ \dots\ q_{i-1}\ \tilde{q}\ q_m\ q_i\ \dots\ q_{m-2}\ p_1\ p_2\ \dots\ p_{i-1}\ p_{m-1}\ p_m\ p_i\ \dots\ p_{m-2} \tag{5.20}$$

or

$$q_1\ q_2\ \dots\ q_{i-1}\ \tilde{q}\ q_m\ q_i\ \dots\ q_{m-2}\ p_1\ p_2\ \dots\ p_{i-1}\ p_m\ p_{m-1}\ p_i\ \dots\ p_{m-2} \tag{5.21}$$

but not both of them are CBOs (since the order of p_{m-1} and p_m is clearly defined). Here, the only differences between the sequences are marked with red color.

1. We assume (5.19) and (5.20) are cyclic base orderings which can be written as edge swap sequences

$$(q_1, p_1), \dots, (q_{i-1}, p_{i-1}), (q_m, p_{m-1}), (\tilde{q}, p_m), (q_i, p_i), \dots, (q_{m-2}, p_{m-2}) \tag{5.22}$$

and

$$(q_1, p_1), \dots, (q_{i-1}, p_{i-1}), (\tilde{q}, p_{m-1}), (q_m, p_m), (q_i, p_i), \dots, (q_{m-2}, p_{m-2})\ . \tag{5.23}$$

Now, we analyze (5.22) and (5.23) with respect to the stronger property of being subsequence-interchangeable assuming that (5.18) is a SIBO for $B^{(m-1)}$. To this end, we consider any (consecutive) subsequence of (5.22) and (5.23) and consider the application of these edge swap sequences to Q. Most of the resulting sets are spanning trees by the assumption that (5.18) is a subsequence-interchangeable base ordering:

(a) Each subsequence of (5.22) and each subsequence of (5.23) which contains *both* red-colored edge swaps is admissible with respect to Q.

(b) Each subsequence of (5.22) and each subsequence of (5.23) which contains *no* red-colored edge swaps is admissible with respect to Q.

(c) Each subsequence of (5.22) which ends with (q_m, p_{m-1}) is admissible with respect to Q (now, the leaf v is joined by p_{m-1} instead of q_m).

(d) Each subsequence of (5.23) which starts with (q_m, p_m) is admissible with respect to Q (now, the leaf v is joined by p_m instead of q_m).

The only subsequences which might not transform Q into another spanning tree of B are

$$(\tilde{q}, p_m), (q_i, p_i), \dots, (q_{i+j-1}, p_{i+j-1}) \tag{5.24}$$

for some $0 \le j \le m-i-1$ with respect to (5.22) and

$$(q_{i-k}, p_{i-k}), \dots, (q_{i-1}, p_{i-1}), (\tilde{q}, p_{m-1}) \tag{5.25}$$

for some $0 \le k \le i-1$ according to (5.23). Choosing $j = 0$ or $k = 0$, this notation means that the subsequences consist only of $(\tilde{q}, p_m)$ or $(\tilde{q}, p_{m-1})$, respectively.

Now, we choose a minimal j and k such that the corresponding subsequences are not admissible. Then, we distinguish two cases.

- If $j > 0$ and $k > 0$ or $j = 0$ and $k = 0$ holds, we obtain a contradiction to our induction hypothesis since exactly one of the sets which we obtain by applying the edge swaps $(\tilde{q}, p_m)$ or $(\tilde{q}, p_{m-1})$ to Q must contain a cycle.
- For both remaining cases, $j = 0$ and $k > 0$ as well as $j > 0$ and $k = 0$, we can construct a bispanning graph such that neither (5.22) nor (5.23) are a SIBO. For $j = 0$ and $k > 0$, we consider the bispanning graph in Figure 5.6. We remove the edge-split according to edge q_m (the edges p_{m-1} and p_m become a new edge $\tilde{p}$) and obtain a bispanning graph $B^{(4)}$ where

  $$(q_1, p_1), (q_2, p_2), (\tilde{q}, \tilde{p}), (q_3, p_3) \tag{5.26}$$

  is a subsequence-interchangeable base ordering of $B^{(4)}$ but both extensions

  $$(q_1, p_1), (q_2, p_2), (q_m, p_{m-1}), (\tilde{q}, p_m), (q_3, p_3)$$

  and

  $$(q_1, p_1), (q_2, p_2), (\tilde{q}, p_{m-1}), (q_m, p_m), (q_3, p_3)$$

  are only cyclic base orderings of $B^{(4)}$ and *not* subsequence-interchangeable. For the case $j > 0$ and $k = 0$, an example is illustrated in Figure 5.7. This example is very similar to the previous one. The sequence

  $$(q_1, p_1), (\tilde{q}, \tilde{p}), (q_2, p_2), (q_3, p_3)$$

  is a subsequence-interchangeable base ordering for the $B^{(4)}$ obtained by reversing the edge-split with respect to q_m. Now, the extensions

  $$(q_1, p_1), (q_m, p_{m-1}), (\tilde{q}, p_m), (q_2, p_2), (q_3, p_3)$$

  and

  $$(q_1, p_1), (\tilde{q}, p_{m-1}), (q_m, p_m), (q_2, p_2), (q_3, p_3)$$

  are only cyclic base orderings of $B^{(4)}$ but *not* subsequence-interchangeable. Nevertheless, it is easy to verify that both examples have several subsequence-interchangeable base orderings: Using another permutation of (5.26) yields a SIBO for the corresponding $B^{(4)}$: if we exchange the edge swap $(\tilde{q}, \tilde{p})$ at the first position, we can apply Theorem 5.7 in order to obtain a SIBO for the given graph.

2. We assume (5.19) and (5.21) are cyclic base orderings. The edge swap notation of (5.21) is

   $$(q_1, p_1), \ldots, (q_{i-1}, p_{i-1}), (\tilde{q}, p_m), (q_m, p_{m-1}), (q_i, p_i), \ldots, (q_{m-2}, p_{m-2}) \;, \tag{5.27}$$

   and, for the sake of completeness, the notation for (5.19) is

   $$(q_1, p_1), \ldots, (q_{i-1}, p_{i-1}), (q_m, p_{m-1}), (\tilde{q}, p_m), (q_i, p_i), \ldots, (q_{m-2}, p_{m-2}) \;. \tag{5.22}$$

   Again, most (consecutive) subsequences of (5.22) and (5.27) have to be admissible with respect to Q by (5.18) being a SIBO for $B^{(m-1)}$:

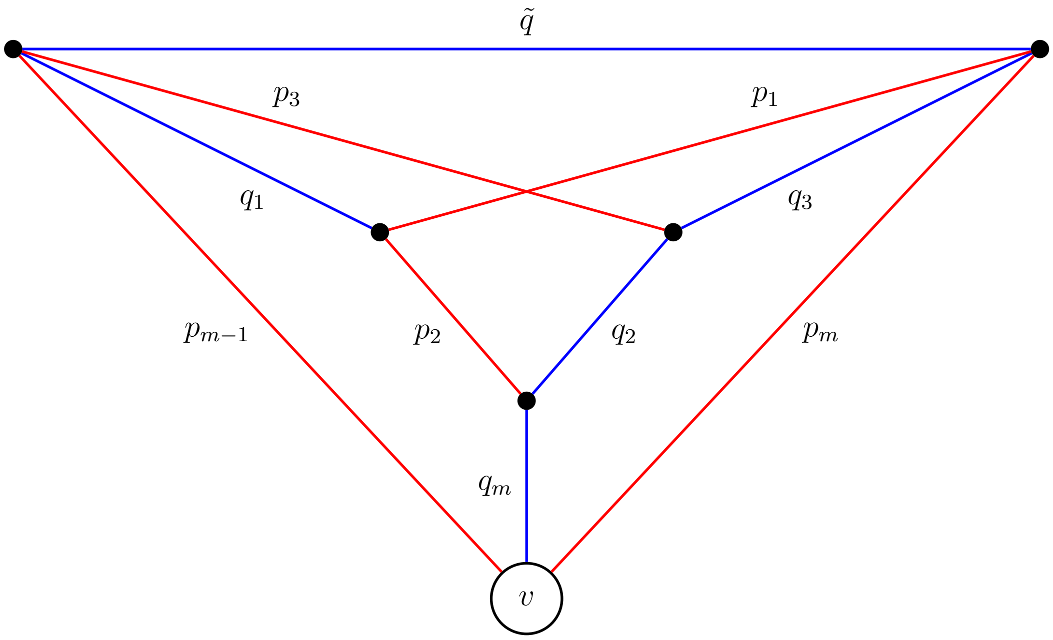


Figure 5.6: Splitting a 'middle' edge (a): Remove the edge-split according to q_m (p_{m-1} and p_m become a new edge $\tilde{p}$). The sequence $(q_1, p_1), (q_2, p_2), (\tilde{q}, \tilde{p}), (q_3, p_3)$ is a SIBO for the resulting graph. Neither $(q_1, p_1), (q_2, p_2), (q_m, p_{m-1}), (\tilde{q}, p_m), (q_3, p_3)$ nor $(q_1, p_1), (q_2, p_2), (\tilde{q}, p_{m-1}), (q_m, p_m), (q_3, p_3)$ is a SIBO for the given graph. (Applying the red-colored edge swaps to Q yields a cycle.)

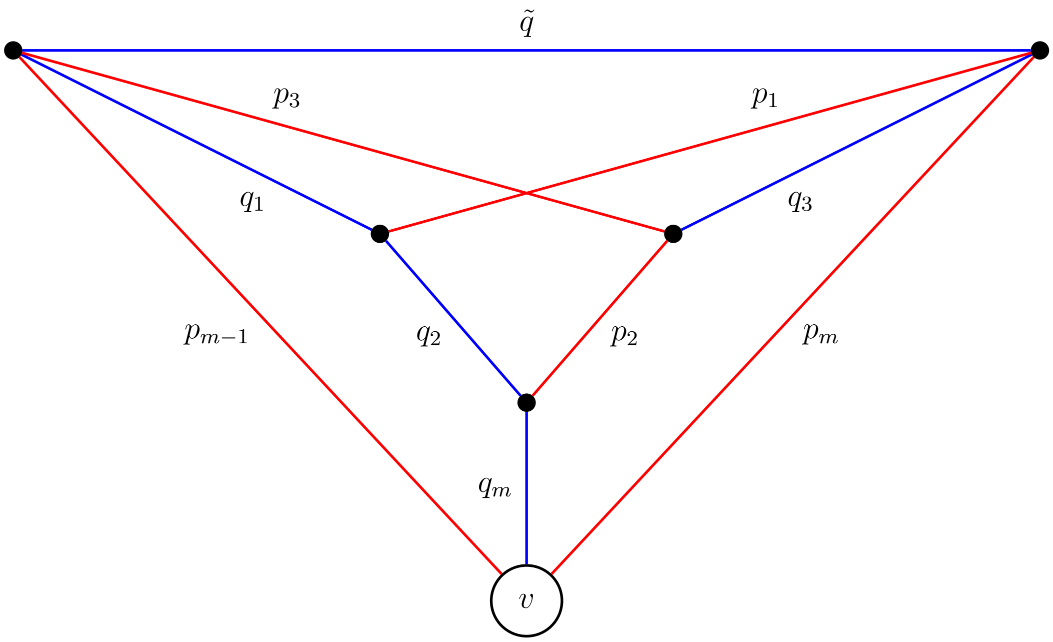


Figure 5.7: Splitting a 'middle' edge (b): Remove the edge-split according to q_m (p_{m-1} and p_m become a new edge $\tilde{p}$). The sequence $(q_1, p_1), (\tilde{q}, \tilde{p}), (q_2, p_2), (q_3, p_3)$ is a SIBO for the resulting graph. Neither $(q_1, p_1), (q_m, p_{m-1}), (\tilde{q}, p_m), (q_2, p_2), (q_3, p_3)$ nor $(q_1, p_1), (\tilde{q}, p_{m-1}), (q_m, p_m), (q_2, p_2), (q_3, p_3)$ is a SIBO for the given graph. (Applying the red-colored edge swaps to Q yields a cycle.)

(a) Each subsequence of (5.22) and each subsequence of (5.27) which contains *both* red-colored edge swaps is admissible with respect to Q.

(b) Each subsequence of (5.22) and each subsequence of (5.27) which contains *no* red-colored edge swaps is admissible with respect to Q.

(c) Each subsequence of (5.22) which ends with (q_m, p_{m-1}) is admissible with respect to Q (now, the leaf v is joined by p_{m-1} instead of q_m).

(d) Each subsequence of (5.27) which starts with (q_m, p_m) is admissible with respect to Q (now, the leaf v is joined by p_m instead of q_m).

But it is possible that a subsequence

$$(\tilde{q}, p_m), (q_i, p_i), \ldots, (q_{i+j-1}, p_{i+j-1}) \tag{5.28}$$

for some $0 \le j \le m-i-1$ is not admissible with respect to Q *and* a subsequence

$$(q_{i-k}, p_{i-k}), \ldots, (q_{i-1}, p_{i-1}), (\tilde{q}, p_m) \tag{5.29}$$

for some $0 \le k \le i-1$ is not admissible with respect to Q. If there is more than one possible value for j or k then we choose minimal ones. In this case, we either have $j = 0$ and $k = 0$ or we have $j > 0$ and $k > 0$ (since the edge swap $(\tilde{q}, p_m)$ appears in both sequences). An example for the first case is illustrated in Figure 5.8. After reversing the edge-split operation according to q_m (the edges p_{m-1} and p_m become a new edge $\tilde{p}$), the resulting bispanning graph $B^{(3)}$ has the subsequence-interchangeable base ordering

$$(q_1, p_1), (\tilde{q}, \tilde{p}), (q_2, p_2)$$

but the extensions

$$(q_1, p_1), (q_m, p_{m-1}), (\tilde{q}, p_m), (q_2, p_2)$$

and

$$(q_1, p_1), (\tilde{q}, p_m), (q_m, p_{m-1}), (q_2, p_2)$$

are only CBOs for the given graph and *not* subsequence-interchangeable. Analogously, Figure 5.9 shows the case $j > 0$ and $k > 0$. Again, we consider the bispanning graph which is obtained by reversing the edge-split according to q_m (the edges p_{m-1} and p_m become a new edge $\tilde{p}$). This graph has the SIBO

$$(q_1, p_1), (q_2, p_2), (\tilde{q}, \tilde{p}), (q_3, p_3), (q_4, p_4)\ .$$

The extensions

$$(q_1, p_1), (q_2, p_2), (q_m, p_m), (\tilde{q}, p_{m-1}), (q_3, p_3), (q_4, p_4)$$

and

$$(q_1, p_1), (\tilde{q}, p_{m-1}), (q_m, p_m), (q_2, p_2), (q_3, p_3), (q_4, p_4)$$

are only CBOs for the given graph but *not* subsequence-interchangeable. Nevertheless, there are SIBOs for both graphs in Figures 5.8 and 5.9.

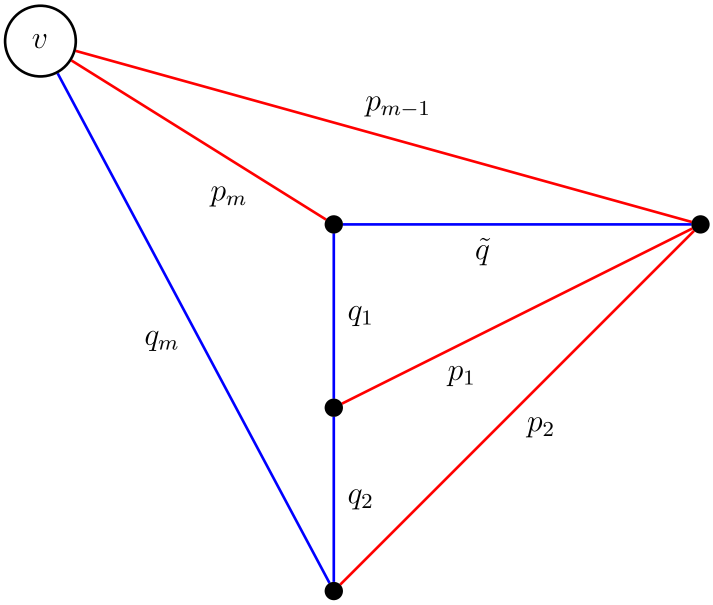


Figure 5.8: Splitting a 'middle' edge (c): Remove the edge-split according to q_m (p_{m-1} and p_m become a new edge $\tilde{p}$). The sequence $(q_1, p_1), (\tilde{q}, \tilde{p}), (q_2, p_2)$ is a SIBO for the resulting graph. Neither $(q_1, p_1), (q_m, p_{m-1}), (\tilde{q}, p_m), (q_2, p_2)$ nor $(q_1, p_1), (\tilde{q}, p_m), (q_m, p_{m-1}), (q_2, p_2)$ is a SIBO for the given graph. (Applying the red-colored edge swaps to Q yields a cycle.)

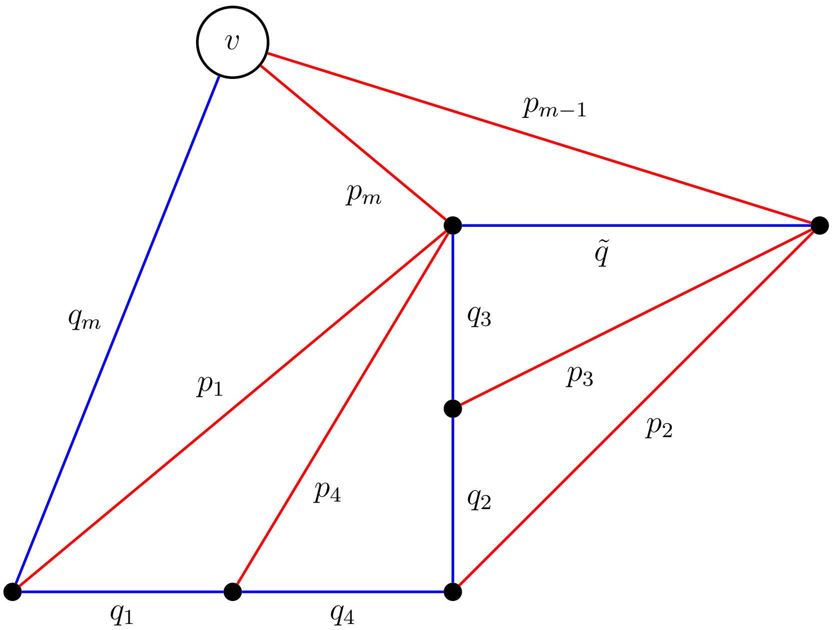
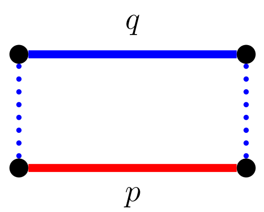


Figure 5.9: Splitting a 'middle' edge (d): Remove the edge-split according to q_m (p_{m-1} and p_m become a new edge $\tilde{p}$). The sequence $(q_1, p_1), (q_2, p_2), (\tilde{q}, \tilde{p}), (q_3, p_3), (q_4, p_4)$ is a SIBO for the resulting graph. Neither $(q_1, p_1), (q_2, p_2), (q_m, p_m), (\tilde{q}, p_{m-1}), (q_3, p_3), (q_4, p_4)$ nor $(q_1, p_1), (q_2, p_2), (\tilde{q}, p_{m-1}), (q_m, p_m), (q_3, p_3), (q_4, p_4)$ is a SIBO for the given graph. (Applying the red-colored edge swaps to Q yields a cycle.)

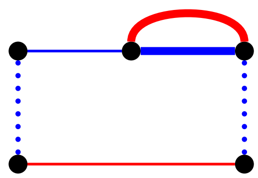
The above examples have the following properties in common: First, we observe that the ideas of the proof of Theorem 5.2 (constructing a CBO) cannot easily be used or extended in order to construct a SIBO. A second property of the given examples is that we only have to permute the edge swaps of the SIBOs containing the edge swap $(\tilde{q}, \tilde{p})$ in such a way that this swap is exchanged at the first or the last position. Afterwards, we can apply Theorem 5.7. Hence, there are two promising possibilities to use the bottom-up approach in order to prove the existence of SIBOs for each bispanning graph.

Splitting edges of boundary edge swaps of a SIBO

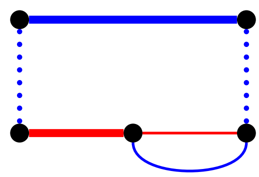
The first possibility is that it might be sufficient to apply edge-splits only to boundary edges (i.e., splitting only a 'first' or 'last' edge). Figure 5.10 shows a graphical representation of the results of Theorem 5.7. Let (q, p) be a boundary edge swap (marked by bold edges). Figure 5.10(b) shows which edges are contained in a boundary edge swap after splitting q. In this figure, these edges are the splitting (red-colored) edge and the edge belonging to the fundamental cycle generated by the splitting edge with respect to Q. Figure 5.10(c) shows which edges are contained in a boundary edge swap after splitting p. Here, these edges are q and the part of p which has vertices in different components of $Q \setminus \{q\}$. Another example is given in Figure 5.11. Starting with a single vertex, we apply two double-leaf attachments that generate two boundary edge swaps (cf. Figure 5.11(c)). Afterwards, we only split edges of boundary edge swaps and we obtain the SIBO $(q_3, p_3), (q_2, p_2), (q_1, p_1), (q_6, p_6), (q_5, p_5), (q_4, p_4)$.



(a) A pair of 'boundary' edges.



(b) Splitting q.



(c) Splitting p.

Figure 5.10: Restricted edge-split operation. The 'boundary' edges are bold.

Transforming a SIBO into another SIBO

The second possibility is to find a relationship between different SIBOs for the same graph. Here, the question is whether or not it is possible to exchange several edge swaps (or only some edges belonging to different edge swaps) in order to transform a SIBO S into another SIBO S' such that a special edge is exchanged within the first (or last) edge swap. In this case, Theorem 5.7 is sufficient. A variant of this is used for the construction of SIBOs for any partition of the wheel graph. In this case, we exploit the fact that a pair of multiple edges (q, p) can be integrated at any position in a given SIBO. We remark that the same property was already used in the analysis of the 'double-leaf attachment' operation.

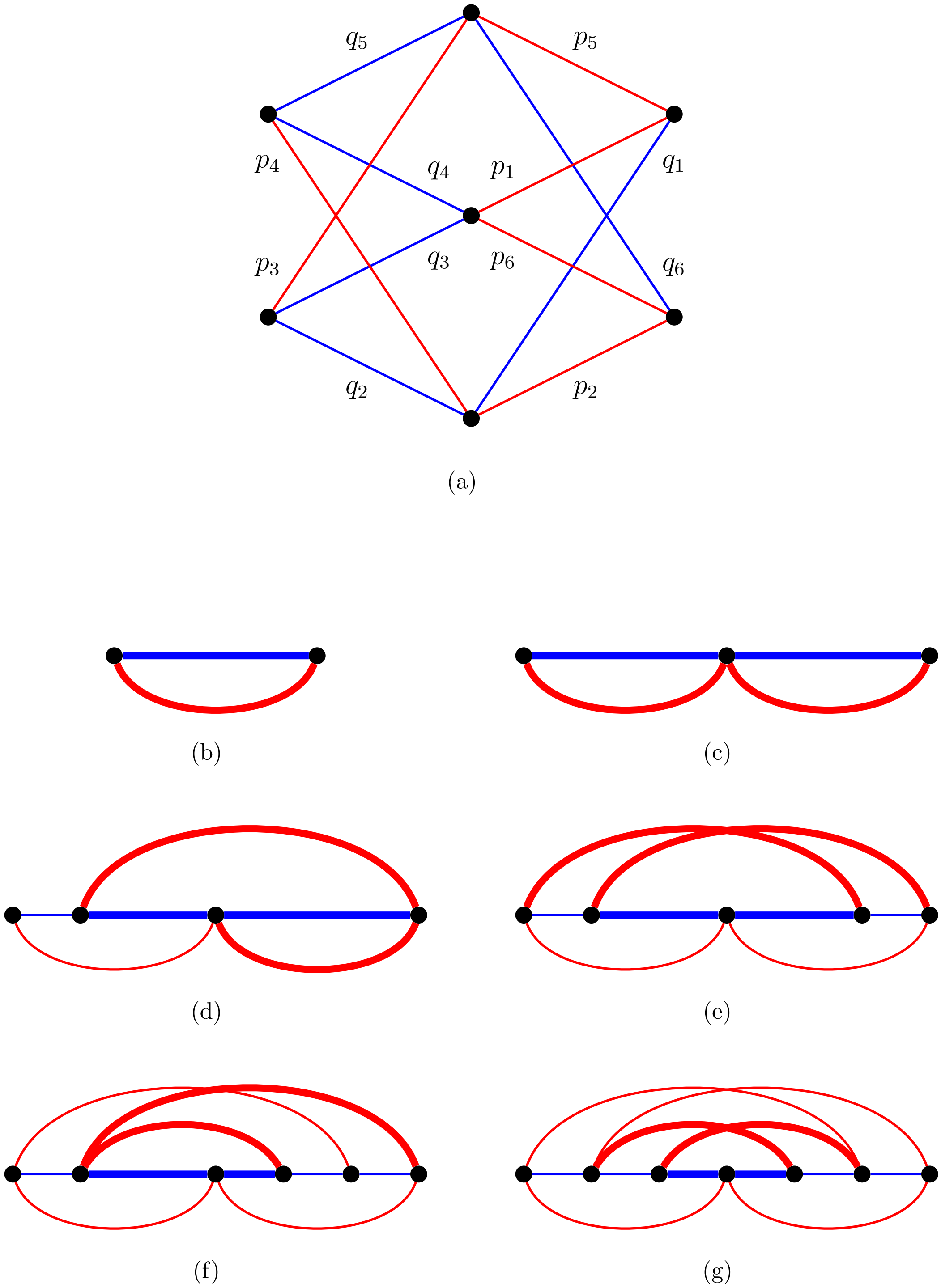


Figure 5.11: A bispanning graph and a construction sequence only consisting of restricted edge-splits or double-leaf attachments. Using Figures (b) through (g), we obtain the SIBO $(q_3, p_3), (q_2, p_2), (q_1, p_1), (q_6, p_6), (q_5, p_5), (q_4, p_4)$.

5.3.2 Top-Down Approach

In this section, we want to break-up a given bispanning graph into smaller components in order to compute a subsequence-interchangeable base ordering for the whole graph. We call this procedure the 'top-down' approach. Most of our ideas follow the same lines as described in Chapter 4. In that chapter, we studied (weighted) bispanning graphs B with edge-connectivity $\lambda(B) = 2$ or vertex-connectivity $\kappa(B) = 1$. Subsequently, we presented a generalization. The crucial observation for this generalization is that in both discussed cases, the considered bispanning graphs are composite, i.e., they contain a non-trivial bispanning subgraph. This fact can also be applied to subsequence-interchangeable base orderings. Here, we can show that if a given bispanning graph B contains a non-trivial bispanning subgraph B_1 then we can decompose B into two smaller bispanning graphs. We can combine SIBOs for each of these graphs in order to obtain a SIBO for the original graph.

Theorem 5.8. *Let $B = (V, P, Q)$ be a bispanning graph and let B_1 be a non-trivial bispanning subgraph of B. Let B_2 be the graph which is obtained from B by contracting the subgraph B_1 to a single vertex. Then, B_2 is a bispanning graph. Furthermore, if B_1 and B_2 have SIBOs S_1 and S_2, respectively, there is a SIBO for B, too.*

Proof. Let $B_1 = (V_1, P_1, Q_1)$ be a non-trivial bispanning subgraph of B and let B_2 be the graph we obtain from B by contracting B_1. We define P_2 and Q_2 to be the remaining edges (after the contraction) of P and Q, respectively. Clearly, if P_2 (or Q_2) is not a spanning tree of B_2 then P (or Q) is not a spanning tree of B, thus, the resulting graph B_2 is indeed bispanning. Furthermore, we observe that each spanning tree of B_1 combined with each spanning tree of B_2 yields a spanning tree of B. Hence, the concatenation of the edge swaps of S_1 and S_2 forms a SIBO of B. □

The essential point in the last proof is that S_1 and S_2 are independent since the spanning trees of B_1 and B_2 can be independently combined. Hence, it is also possible to interleave S_1 and S_2 such that the internal order of the elements with respect to S_1 and S_2 remains unchanged. The number of different SIBOs is an easy combinatorial task. More precisely, let m_1 and m_2 be the length of S_1 and S_2, respectively. Then, the number of possible SIBOs for B is larger by a factor of

$$\binom{m_1 + m_2}{m_1}$$

than the number of SIBOs for B_1 times the number of SIBOs for B_2.

Given an arbitrary bispanning graph, we can recursively apply Theorem 5.8 until the considered graphs become atomic. Hence, we can formulate the following corollary.

Corollary 5.9. *If there exists a subsequence-interchangeable base ordering for each atomic bispanning graph $B = (V, P, Q)$ then there exists a SIBO for each bispanning graph.*

5.3.3 The Class of Wheel Graphs

As seen above, we only have to focus on atomic bispanning graphs. It is easy to construct numerous atomic bispanning graphs only by using the results of Theorems 5.6 and 5.7. Unfortunately, we do not know whether or not these theorems are sufficient to construct all bispanning graphs. A whole graph class of atomic bispanning graphs for which we can construct SIBOs are the wheel graphs.

Definition 5.10. *The wheel on $n \geq 4$ vertices, denoted by W_n, is a graph consisting of a cycle of $n-1$ vertices where each of these vertices is connected to a special vertex, the so-called hub. The edges incident to the hub are the spokes of the wheel.*

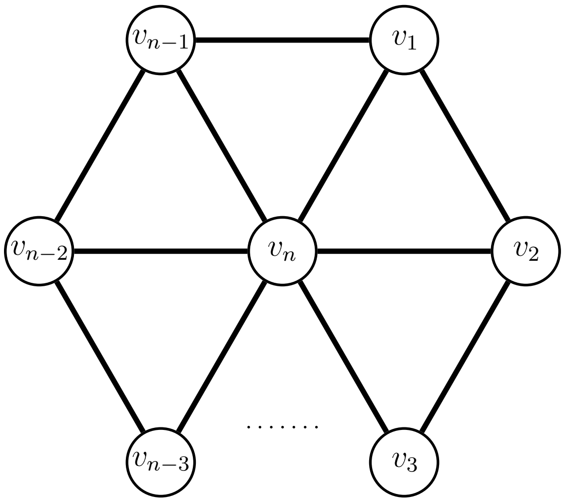


Figure 5.12: The wheel W_n.

Theorem 5.11. *Let $W_n = (V, E)$ be the wheel on n vertices and let $E = P \cup Q$ be an arbitrary partition of its edges into two spanning trees P and Q. Then, there is a subsequence-interchangeable base ordering for the bispanning graph $B = (V, P, Q)$.*

Proof. First, we observe that each of the cycle vertices v_i with $1 \leq i < n$ is either incident to two edges of P or incident to two edges of Q. We claim that there exists at least one index $1 \leq i < n$ such that v_i's spoke and one of its incident cycle edges belong either to P or to Q. Otherwise, we obtain a contradiction to P and Q being spanning trees. Relabel any vertex having this property (the spoke and one cycle edge belong to the same tree) with v_1. Without loss of generality, we assume v_1 is connected to the hub v_n of W_n by an edge belonging to P. Let v_2 be the cycle vertex which is connected to v_1 by an edge belonging to P, too. We relabel the remaining vertices with $v_3, \ldots, v_{n-1}$ such that v_3 is a neighbor of v_2, v_4 is a neighbor of v_3, and so on.

Let $q_m = \{v_1, v_{n-1}\}$, $p_{m-1} = \{v_1, v_2\}$, and $p_m = \{v_1, v_n\}$. Analogous to the proof of Theorem 5.2, we remove the vertex v_1 and its incident edges and introduce a new edge $\tilde{p} = \{v_2, v_n\}$, thus, we reverse the edge-split with respect to q_m. Let $B^{(n-1)} = (V \setminus \{v_1\}, P \setminus \{p_{m-1}, p_m\} \cup \{\tilde{p}\}, Q \setminus \{q_m\})$ with $\tilde{p} = \{v_2, v_n\}$ be the remaining bispanning graph. Let $\tilde{q}$ be the edge connecting v_2 with v_n in W_n. Obviously, $\tilde{p}$ and $\tilde{q}$ form a pair of multiple edges. Suppose we are able to construct the following SIBO of $B^{(n-1)}$

$$(q_1, p_1), \ldots, (\tilde{q}, \tilde{p}), \ldots, (q_{m-2}, p_{m-2}) \; . \tag{5.30}$$

How to construct such a subsequence-interchangeable base ordering is rather simple and we explain it in detail later. Since $\tilde{q}$ and $\tilde{p}$ are a pair of multiple edges, they have to be exchanged by each other. Moreover, we observe that the edge swap $(\tilde{q}, \tilde{p})$ can be moved to each position in (5.30) without changing the order of the remaining edge swaps in order to preserve the property of being a subsequence-interchangeable base ordering. In particular, the sequences

$$(\tilde{q}, \tilde{p}), (q_1, p_1), \ldots, (q_{m-2}, p_{m-2})$$

and

$$(q_1, p_1), \ldots, (q_{m-2}, p_{m-2}), (\tilde{q}, \tilde{p})$$

are SIBOs. Now we are ready to apply Theorem 5.7 in order to obtain a subsequence-interchangeable base ordering for W_n (or B).

To conclude the proof, we have to show how to construct the SIBO (5.30) for $B^{(n-1)}$. Note that $(\tilde{q}, \tilde{p})$ is a pair of multiple edges in $B^{(n-1)}$. Hence, using the ideas of the previous section (Theorem 5.8), we can contract them. Again, a new pair of multiple edges emerges. It is easy to see that this situation will occur each time when contracting a multiple edge until the graph is exhausted, i.e., it contains only a single vertex. In Figure 5.13, this situation is illustrated for the wheel W_7. We can even stop this procedure if the graph consists of a pair of multiple edges. Since the exchange of multiple edges is clearly defined, the following sequence of edge swaps is a SIBO for $B^{(n-1)}$

$$(\tilde{q}, \tilde{p}), (\{v_2, v_3\}, \{v_3, v_n\}), (\{v_3, v_4\}, \{v_4, v_n\}), \ldots, (\{v_{n-2}, v_{n-1}\}, \{v_{n-1}, v_n\}) \; . \tag{5.31}$$

This proves the theorem. □

Note that in (5.31), the order of each edge swap is defined by the partition of $B^{(n-1)}$ into P and Q. For example, the edge swap $(\{v_2, v_3\}, \{v_3, v_n\})$ must be changed into $(\{v_3, v_n\}, \{v_2, v_3\})$ if $\{v_3, v_n\}$ belongs to Q. Clearly, both edges of any edge swap in (5.31) do not belong to the same spanning tree since otherwise we get a pair of multiple edges belonging either to Q or to P.

From the proof of Theorem 5.11, we immediately get Algorithm 2 for computing a SIBO of any partition of the W_n. This algorithm computes a SIBO in time $\mathcal{O}(n)$ without performing any contractions.

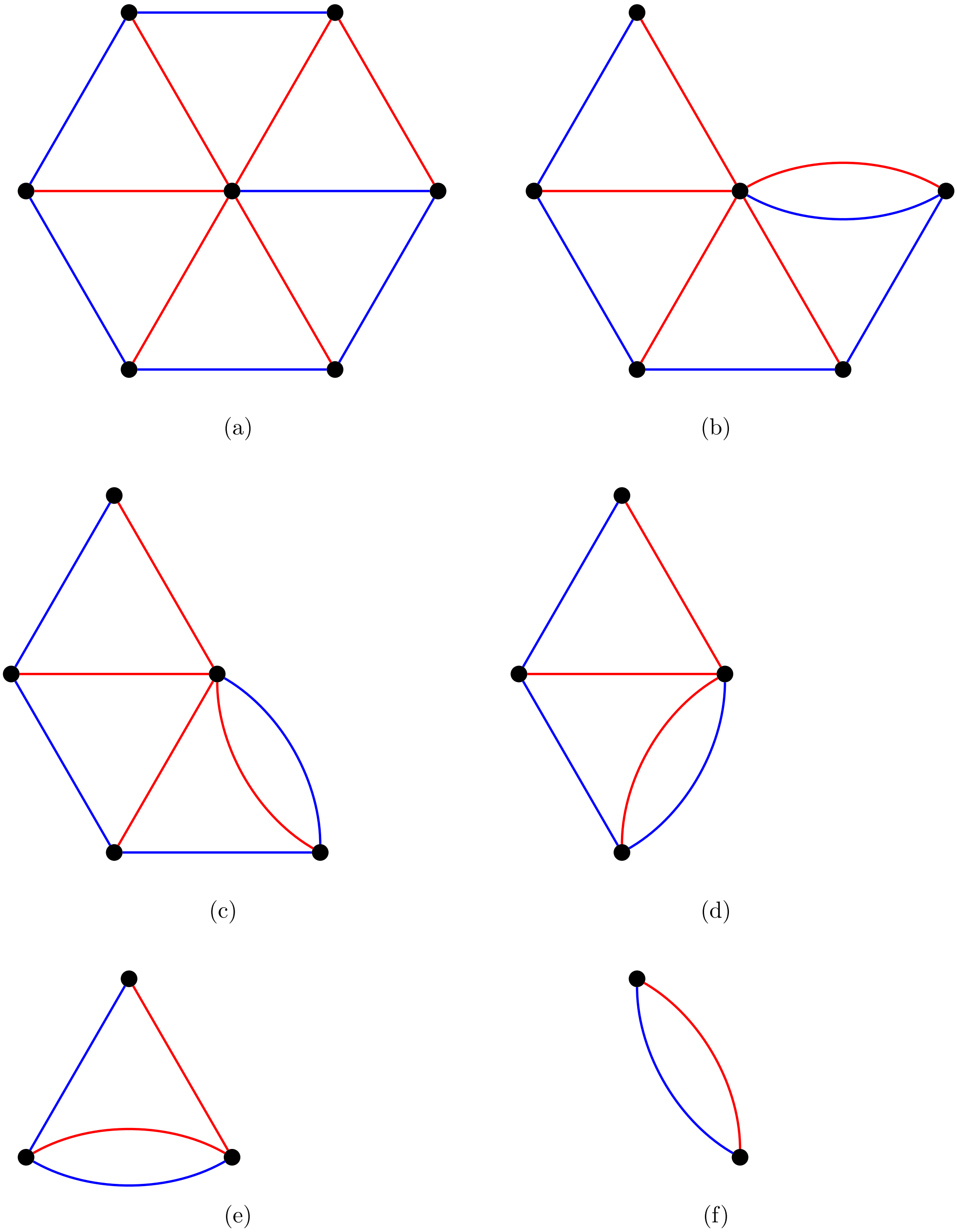


Figure 5.13: Processing a partition of the wheel W_7.

Algorithm 2: SIBO($W_n = (V, P, Q)$)

Input: the wheel W_n and with 'P' or 'Q' labeled edges
Output: a SIBO S for W_n with respect to Q

begin
  $S =$ `null`
  compute the hub v_n
  compute $v_1 \in V \setminus \{v_n\}$ such that `label`$[\{v_1, x\}] \neq$ `label`$[\{v_1, y\}]$ and $x, y \neq v_n$
  if `label`$[\{v_1, x\}] ==$ `label`$[\{v_1, v_n\}]$ **then**
    label x with v_2
  else
    label y with v_2
  label the cycle vertices according to the order of v_1 and v_2 with $v_3, \ldots, v_{n-1}$
  for $i = 3$ **to** $n - 1$ **do**
    if `label`$[\{v_i, v_n\}] == Q$ **then**
      $S = S \circ (\{v_i, v_n\}, \{v_i, v_{i-1}\})$
    else
      $S = S \circ (\{v_i, v_{i-1}\}, \{v_i, v_n\})$
  if `label`$[\{v_1, v_2\}] == Q$ **then**
    `/* Splitting an edge of Q */`
    if $\{v_1, v_n\} \in C(Q, \{v_1, v_{n-1}\})$ **then**
      $S = (\{v_1, v_2\}, \{v_2, v_n\}) \circ S$
      $S = (\{v_1, v_n\}, \{v_1, v_{n-1}\}) \circ S$
    else
      $S = (\{v_1, v_n\}, \{v_2, v_n\}) \circ S$
      $S = (\{v_1, v_2\}, \{v_1, v_{n-1}\}) \circ S$
  else
    `/* Splitting an edge of P */`
    if $\{v_2, v_n\} \in C(Q, \{v_1, v_2\})$ **then**
      $S = (\{v_1, v_{n-1}\}, \{v_1, v_n\}) \circ S$
      $S = (\{v_2, v_n\}, \{v_1, v_2\}) \circ S$
    else
      $S = (\{v_1, v_{n-1}\}, \{v_1, v_2\}) \circ S$
      $S = (\{v_2, v_n\}, \{v_1, v_n\}) \circ S$
end

5.4 Concluding Remarks

In this chapter, we introduced a new kind of base ordering restricted to spanning trees of bispanning graphs. The orderings which fulfill the stronger property of being subsequence-interchangeable are a subclass of the known cyclic base orderings. More precisely, we have shown the relationship:

$$\mathcal{SIBO} \subseteq \mathcal{CBO} \subseteq \mathcal{PQ} \subseteq \mathcal{BO}$$

where $\mathcal{BO}$ denotes the set of all base orderings and $\mathcal{PQ}$ denotes the set of all paths in the tree graph between the corresponding bases or spanning trees.

We remark that our results can be extended to general graphs (and spanning trees). If both spanning trees are edge-disjoint and the given graph has further edges then we consider only the bispanning subgraph consisting of these two spanning trees. Otherwise, i.e., if both spanning trees have common edges, we do not need to change these edges (or exchange them by themselves). Hence, (cyclic as well as subsequence-interchangeable) base orderings can be extended to general graphs.

We discussed several approaches in order to find a proof for the existence (and an algorithm for the construction) of subsequence-interchangeable base orderings for each partition of a bispannable graph into two spanning trees. From our point of view, the most promising approach is to further study transformations between different SIBOs of a bispanning graph. We have already applied such transformations in their simplest form to wheel graphs: the position of the edge swap (q, p) of a pair of multiple edges can be arbitrarily exchanged without destroying the property of being subsequence-interchangeable. Furthermore, it is important to show or to disprove that the restricted edge-split operation and the double-leaf attachment operation are sufficient to construct SIBOs.

We have seen that it is sufficient to look for a construction algorithm of SIBOs only for *atomic* bispanning graphs, i.e., bispanning graphs that have a minor isomorphic to K_4. In Chapter 4, atomic bispanning graphs are also the crucial point. In fact, there is a deeper correlation between these problems. In the following, we describe this stronger connection and point out why it might be important to prove the existence of subsequence-interchangeable base orderings. To this end, we recall the topic of the last chapter.

In Chapter 4, we proposed a conjecture regarding weighted bispanning graphs. More precisely, let $B = (V, P, Q)$ be a weighted bispanning graph such that Q is the only spanning tree with weight $w(Q)$ and P has weight strictly smaller than Q. We conjectured that given such a bispanning graph, there are at least $n-1$ distinct spanning trees with pairwise different weights strictly smaller than $w(Q)$. Moreover, we conjectured that it suffices to count only so-called partition spanning trees. Now, the connection between SIBOs and the problem of counting spanning trees of distinct weights is summarized in the following proposition.

Proposition 5.12. *Let $B = (V, P, Q, w)$ be a weighted bispanning graph such that Q has unique weight regarding all spanning trees of B, i.e., we have $\sigma(B, Q) = 1$. Let S be a SIBO for B (with respect to Q). Then, the spanning trees lying on the path (corresponding to S) between Q and P have pairwise different weights.*

Proof. Let $B = (V, P, Q, w)$ be a weighted bispanning graph and let $S = (s_1, \ldots, s_m)$ be a SIBO for B. We suppose that there are two spanning trees T_i and T_j with

$$T_i = (s_1 \circ \ldots \circ s_i)(Q) \qquad \text{and} \qquad T_j = (s_1 \circ \ldots \circ s_j)(Q)$$

for distinct values of $i, j \in \{1, \ldots, m\}$ such that $w(T_i) = w(T_j)$. Without loss of generality, we assume $i < j$. Hence, applying the edge swaps s_{i+1} through s_j to T_i does not change the spanning tree weight. Since the subsequence $S' = (s_{i+1}, \ldots, s_j)$ is also admissible with respect to Q, we obtain a spanning tree $T' = (s_{i+1} \circ \ldots \circ s_j)(Q)$ that has weight $w(Q)$ in contradiction to $\sigma(B, Q) = 1$. $\square$

We observe that each spanning tree of such a path between Q and P consists only of partition spanning trees. Hence, the complement of each spanning tree is a spanning tree, too. Then, it easily follows by the pigeon-hole principle that at least $n-1$ spanning trees (also counting the complementary spanning trees) of such a path have a weight less than $w(Q)$. Unfortunately, the spanning tree weights of such a path and the weights of the complementary spanning trees do not have to be distinct. For example, we consider the partition of the K_4 given in Figure 4.9 on Page 55 and introduce a weight function $w \colon E \to \mathbb{N}$ with

$$w(a) = 3 \qquad w(b) = 8 \qquad w(c) = 3 \qquad w(d) = 1 \qquad w(e) = 6 \qquad w(f) = 6\,. \tag{5.32}$$

Now, we consider the first and the third path in Figure 5.2 on Page 66. Because of symmetry, we can omit the second and the fourth path. The spanning tree weights according to the first path are

$$w(\{a, b, c\}) = 14 \qquad w(\{b, c, d\}) = 12 \qquad w(\{b, d, e\}) = 15 \qquad w(\{d, e, f\}) = 13\,.$$

The complementary spanning tree weights of this path are

$$w(\{d, e, f\}) = 13 \qquad w(\{a, e, f\}) = 15 \qquad w(\{a, c, f\}) = 12 \qquad w(\{a, b, c\}) = 14\,.$$

Hence, this example shows that all spanning trees (including the complementary spanning trees) of a path corresponding to a SIBO do not have to be distinct. Unfortunately, the spanning tree weights of the other path (the third path according to Figure 5.2) are even worse. Here, the weights are

$$w(\{a, b, c\}) = 14 \qquad w(\{a, c, e\}) = 12 \qquad w(\{c, e, f\}) = 15 \qquad w(\{d, e, f\}) = 13$$

and the complementary weights are

$$w(\{d, e, f\}) = 13 \qquad w(\{b, d, f\}) = 15 \qquad w(\{a, b, d\}) = 12 \qquad w(\{a, b, c\}) = 14\,.$$

Hence, given the weight function (5.32), it is not sufficient to count only spanning trees lying on a path corresponding to a subsequence-interchangeable base ordering. In general, we only have at least $\lceil n/2 \rceil$ distinct weights. But there is a gleam of hope: We observe that the spanning tree weights (and the edge weights according to 5.32) are 'symmetric' (e.g., we have $w(a) = w(c)$ and $w(e) = w(f)$) which results in $\sigma(B, P) = \sigma(B, Q) = 1$, i.e., the spanning tree P also has unique weight. But this case was already solved in Chapter 3. Hence, if the weight function becomes 'asymmetric', i.e., we have $\sigma(B, P) > 1$, it might be impossible that all weights of all SIBOs become symmetric as in the example above.

Now, look at the weight function of the K_4 given in Figure 4.1, which is

$$w(a) = 4 \qquad w(b) = 5 \qquad w(c) = 3 \qquad w(d) = 6 \qquad w(e) = 2 \qquad w(f) = 3\ . \quad (5.33)$$

As already mentioned in Chapter 4, this weight function implies $\sigma(B, Q) = 1$ and $\sigma(B, P) = 4$. Then, the spanning trees (and their complements) of the first path in Figure 5.2 are sufficient. More precisely, the spanning tree weights are

$$w(\{a, b, c\}) = 12 \qquad w(\{b, c, d\}) = 14 \qquad w(\{b, d, e\}) = 13 \qquad w(\{d, e, f\}) = 11$$

and the complementary weights are

$$w(\{d, e, f\}) = 11 \qquad w(\{a, e, f\}) = 9 \qquad w(\{a, c, f\}) = 10 \qquad w(\{a, b, c\}) = 12$$

implying $\mathrm{ord}(B, Q) \geq \mathrm{ord}'(B, Q) \geq 4$. Unfortunately, it is difficult to find weight functions satisfying $\sigma(B, P) > 1$ *and* $\sigma(B, Q) = 1$ for large (atomic) bispanning graphs B.

The above observations show the connection of subsequence-interchangeable base orderings to the problem of counting (partition) spanning trees with distinct weight in Chapters 3 and 4. We have seen that the crucial point is to analyze atomic (weighted) bispanning graphs. This graph class has the property of containing a minor isomorphic to K_4. In Chapter 3, we have seen that the cycle matroid of a graph is strongly base orderable if the graph does not contain such a minor. For these graphs, the existence of a subsequence-interchangeable base ordering easily follows by Definition 3.10.

Chapter 6

Graph-Approximating Spanning Trees

6.1 Introduction

In this chapter, we consider a problem, which is related to the simplification of graphs with respect to the number of edges. Problems of this kind can be summarized by the term *network sparsification*. The intention is to thin out the graph while retaining certain network characteristics, e.g., the distances between node pairs. The aim of this task is to reduce the complexity of a given graph in order to simplify computations of network problems or to feature a concise visualization of a complex network with its most important structural properties. For example, the network can be made more amenable to visual examination.

Carrying this sparsification to an extreme, the resulting graph is required to be a spanning tree since the elements of this graph class have a minimum number of edges among all connected subgraphs. Furthermore, spanning trees offer a variety of useful properties which can be exploited for fast algorithms even if the considered problems are (in general) $\mathcal{NP}$-hard. Problems of this kind are also known as *tree spanner* problems.

In this chapter, we consider the problem of computing a spanning tree of a graph, that minimizes, in its simplest form, the sum of the distances between all pairs of nodes, that were connected by an edge in the original graph. Actually, we consider a more general form, where the sum is computed of pth powers of the respective distances (or distance differences), i.e., the calculation is made with respect to the L_p-norm. We call our problem GAST.

The problem is related to a couple of other problems. On the one hand, it is similar to the problem of computing distance-minimizing or distance-approximating spanning trees (DMST, DAST, [EKM+08]). In contrast to the setting in this chapter, the DMST and DAST problems consider the distances of *all* vertex pairs (instead of only pairs connected by single edges in the original graph). Both problems, DMST and DAST, were shown to be $\mathcal{NP}$-complete for all norms L_p, $p \in \mathbb{N}$. On the other hand, GAST is related to the problem of computing a minimum fundamental cycle basis (Min-FCB) for a weighted undirected graph. Here, the aim is to compute a cycle basis (a spanning tree, respectively) that causes a minimum sum of the weights of all fundamental cycles. This problem is also known to be $\mathcal{NP}$-complete [DPK82]. Galbiati and Amaldi [GA04] proposed an approximation algorithm achieving an approximation ratio of $2^{\mathcal{O}(\sqrt{\log n \log\log n})}$. Their approach used a related problem introduced by Hu, namely the Minimum Communication Cost Spanning Tree Problem [Hu74], which was shown to be approximable within the same factor by Peleg and Reshef [PR98]. Elkin, Emek, Spielman, and Teng improved this ratio to $\mathcal{O}(\log^2 n \log\log n)$ [EEST08]. For the simplest form of our problem, i.e., GAST with respect to the L_1-norm, we give a reduction to Min-FCB implying the same approximation ratio.

6.2 The 2-Hitting-Set Gadget

For the reduction, which we accomplish in the next section, we need the Vertex Cover problem. This problem is well known to be $\mathcal{NP}$-complete [GJ79]. To avoid confusion between 'vertices' and 'edges' of the instance of Vertex Cover and of the constructed graph, we use the less common terminology of the equivalent 2-Hitting Set (2HS) problem, i.e., we use 'literals' (vertices) and 'clauses' (edges). The problem is formally defined as follows:

Problem:	2-Hitting Set (2HS).
Input:	A triple $(\mathcal{C}, \mathcal{S}, k)$ consisting of a family $\mathcal{C} = \{C_1, \dots, C_m\}$ of 2-element subsets of a set $\mathcal{S} = \{s_1, \dots, s_n\}$ and a number $k \in \{1, \dots, n\}$.
Question:	Is there a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $\lvert\mathcal{S}'\rvert \leq k$ and the set $C_\mu \cap \mathcal{S}'$ is not empty for each $\mu \in \{1, \dots, m\}$?

A subset $\mathcal{S}' \subseteq \mathcal{S}$ having the required properties is called an *admissible solution* to a 2HS instance $(\mathcal{C}, \mathcal{S}, k)$. For a given 2HS instance, we define the graph $G(\mathcal{C}, \mathcal{S})$ (a slight simplification to [EKM+08]) as follows:

- For each $s_\mu \in \mathcal{S}$, $\mu \in \{1, \dots, n\}$, we define a *literal gadget* G_μ consisting of two *connection vertices* v_μ and v'_μ. Both vertices are connected by the so-called *elongation path* $(v_\mu, e_1^\mu, \dots, e_{m+1}^\mu, v'_\mu)$ of length $m+2$ and the so-called *literal path* $(v_\mu, l_1^\mu, \dots, l_m^\mu, v'_\mu)$ of length $m+1$.

- For each $\mu \in \{1, \dots, n-1\}$, we connect the literal gadgets G_μ and $G_{\mu+1}$ by adding an edge $\{v'_\mu, v_{\mu+1}\}$.

- Additionally, we introduce a vertex v'_0 which is connected to the first literal gadget G_1 by the edge $\{v'_0, v_1\}$.

- For each $\mathcal{C}_\mu = \{s_\nu, s_\kappa\}$, we define a *clause path* of length $2n(m+2)$ that connects the vertices l_μ^ν and l_μ^κ and a *safety path* of length $2n(m+2)$ that connects the vertices v_0' and l_μ^ν whereas we assume without loss of generality that $\nu < \kappa$.

In Figure 6.1, there is an illustration of a 2HS graph representation $G(\mathcal{C}, \mathcal{S})$ where $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$ and $\mathcal{C} = \{\{s_1, s_3\}, \{s_2, s_4\}, \{s_1, s_4\}, \{s_3, s_4\}\}$.

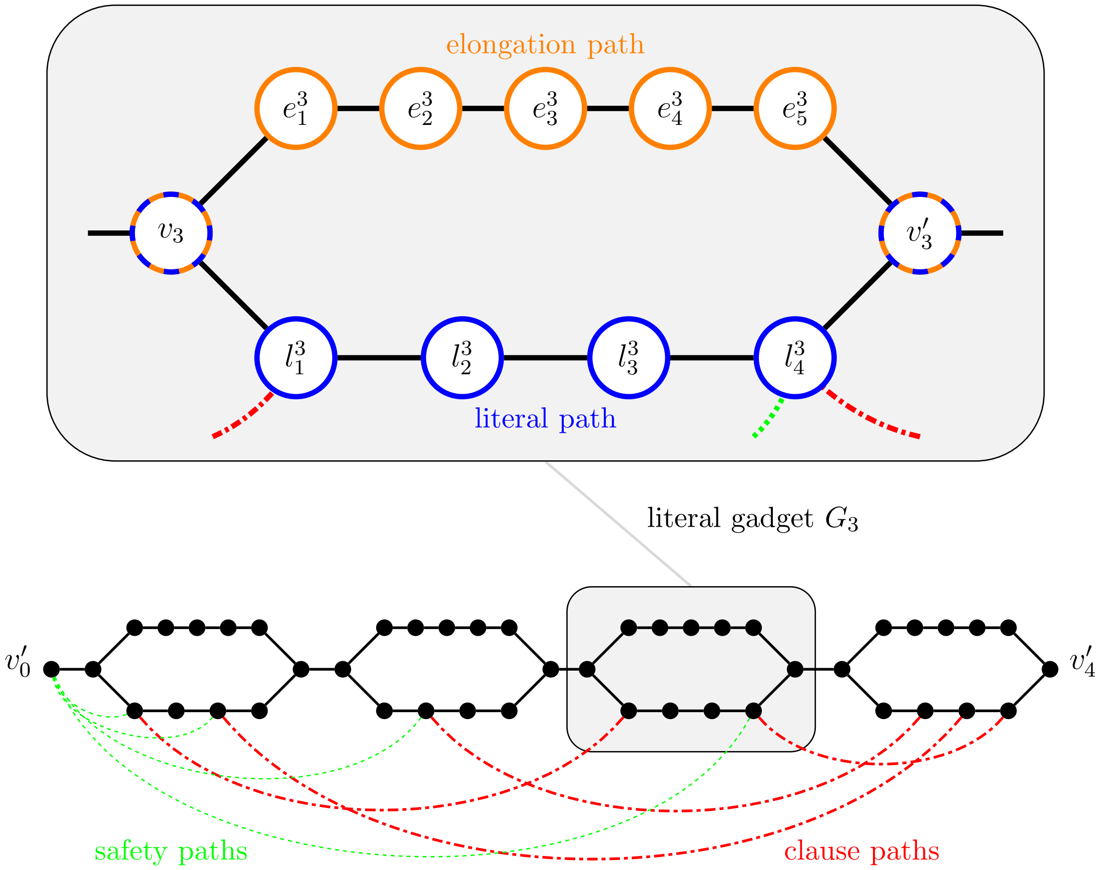


Figure 6.1: Graph representation $G(\mathcal{C}, \mathcal{S})$ of a 2HS instance.

The following lemma is a slight modification of a lemma in [EKM+08].

Lemma 6.1. *Let* $(\mathcal{C}, \mathcal{S}, k)$ *be an instance of* 2HS. *Then, we have*

$$d_{G(\mathcal{C},\mathcal{S})}(v_0', v_n') = n(m+2) \; .$$

Moreover, there exists an admissible solution $\mathcal{S}' \subseteq \mathcal{S}$ *of size* $|\mathcal{S}'| \leq k$ *if and only if there exists a spanning tree* T *of* $G(\mathcal{C}, \mathcal{S})$ *containing all edges in the clause paths such that*

$$d_T(v_0', v_n') \leq d_{G(\mathcal{C},\mathcal{S})}(v_0', v_n') + k \; .$$

Proof. First, we observe that any path from v'_0 to v'_n using a clause or safety path has length at least $2n(m+2)$ whereas the shortest path between v'_0 and v'_n via literal paths has length $n(m+2)$. Thus, it holds that $d_{G(\mathcal{C},\mathcal{S})}(v'_0, v'_n) = n(m+2)$. For the second statement, we consider both directions separately. We start by proving the 'only if' direction. To this end, let $\mathcal{S}'$ be an admissible solution to the 2HS instance $(\mathcal{C}, \mathcal{S}, k)$. We construct a spanning tree of the graph representation $G(\mathcal{C}, \mathcal{S})$ as follows:

1. For each $s_\mu \in \mathcal{S}'$, we remove the edge $\{l^\mu_m, v'_\mu\}$ which is the last edge on the literal path of the literal gadget G_μ.

2. For each $s_\mu \notin \mathcal{S}'$, we remove the edge $\{v_\mu, e^\mu_1\}$ which is the first edge on the elongation path of the literal gadget G_μ.

3. For each $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$ do the following: if $s_\nu \in \mathcal{S}'$ then remove the edge $\{l^\nu_{\mu-1}, l^\nu_\mu\}$. If $s_\kappa \in \mathcal{S}'$ then remove the edge $\{l^\kappa_{\mu-1}, l^\kappa_\mu\}$. Here, we denoted $l^\nu_0 = v_\nu$ and $l^\kappa_0 = v_\kappa$. If not both s_ν and s_κ are elements of $\mathcal{S}'$ then remove an arbitrary edge from the safety path between v'_0 and l^ν_μ.

Note that no edge from a clause path was removed during this construction. Now, we have to prove that each cycle in $G(\mathcal{C}, \mathcal{S})$ is broken when applying the three construction rules. By the first and second construction rule, at least one edge of each cycle induced by the literal and elongation paths is removed. The cycles induced by the clause and safety paths are broken by the first and third construction rule: For each clause $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$, at least one of the sets $\{\{l^\nu_{\mu-1}, l^\nu_\mu\}, \{l^\nu_m, v'_\nu\}\}$ and $\{\{l^\kappa_{\mu-1}, l^\kappa_\mu\}, \{l^\kappa_m, v'_\kappa\}\}$ is removed because $\mathcal{S}'$ is an admissible solution. Thus, either l^ν_μ or l^κ_μ is not reachable via the clause path from v_ν or v'_ν (v_κ or v'_κ, respectively). An edge from the safety path is removed, except if both $\{\{l^\nu_{\mu-1}, l^\nu_\mu\}, \{l^\nu_m, v'_\nu\}\}$ and $\{\{l^\kappa_{\mu-1}, l^\kappa_\mu\}, \{l^\kappa_m, v'_\kappa\}\}$ are removed (third construction rule), in which case neither l^ν_μ nor l^κ_μ is reachable via the clause path from any vertex $v_\nu, v'_\nu, v_\kappa, v'_\kappa$.

A cycle induced by multiple clause paths not going through any connection vertices cannot occur since the connection is broken at one of the literals in $\mathcal{S}'$. As a result, the path between v'_0 and v'_n in T is leading through elongation ($s_\mu \in \mathcal{S}'$) or literal ($s_\mu \notin \mathcal{S}'$) paths only, and does not contain any safety or clause path.

By the construction of the graph representation $G(\mathcal{C}, \mathcal{S})$, the distance of v_μ and v'_μ via a literal path is shorter by one compared to the distance via an elongation path. Thus, it holds that

$$\begin{aligned} d_T(v'_0, v'_n) &= (n - |\mathcal{S}'|)(m+2) + |\mathcal{S}'|(m+3) = n(m+2) + |\mathcal{S}'| \\ &\leq d_{G(\mathcal{C},\mathcal{S})}(v'_0, v'_n) + k \ . \end{aligned}$$

To establish the 'if' direction, let T be a spanning tree of $G(\mathcal{C}, \mathcal{S})$ containing all clause path edges and satisfying $d_T(v'_0, v'_n) \leq d_{G(\mathcal{C},\mathcal{S})}(v'_0, v'_n) + k$. The path between v'_0 and v'_n in T cannot lead through clause or safety paths since

$$d_{G(\mathcal{C},\mathcal{S})}(v'_0, v'_n) + k \leq n(m+3) < 2n(m+2) \ .$$

Hence, this path contains only literal and/or elongation paths.

By construction, the length of any (intact) elongation path is $m + 2$, the length of any (intact) literal path is $m + 1$. Therefore, the path from v'_0 to v'_n contains exactly k elongation paths. Let $\mathcal{S}'$ be the set of literals s_μ for which this path leads from v_μ to v'_μ via an elongation path. Here, the literal path must be broken since otherwise T is not a spanning tree. Conversely, for every $s_\mu \notin \mathcal{S}'$, the literal path is not broken, i.e., $(v_\mu, l^\mu_1, \dots, l^\mu_m, v'_\mu)$ is a path in T. Assume, there is a clause $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$ (where $\nu < \kappa$) such that $C_\mu \cap \mathcal{S}' = \emptyset$. The clause path corresponding to the clause C_μ connects l^ν_μ with l^κ_μ. Since $s_\nu, s_\kappa \notin \mathcal{S}'$, the vertex l^ν_μ is connected to v'_ν which is connected to v^κ which is connected to l^κ_μ. Hence, we obtain a cycle in contradiction to T being a spanning tree. $\square$

6.3 Graph-Approximating Spanning Trees

In this section, we consider the problem of computing a spanning tree of a graph G that minimizes the distances of pairs of vertices which are connected by an edge in the original graph. We study this problem under certain matrix norms. First, we need further definitions and notations. Let $G = (V, E)$ be an undirected simple graph. The adjacency matrix of G is a matrix $A_G \in \{0, 1\}^{n \times n}$ such that $A_G[i, j] = 1$ if and only if $\{v_i, v_j\} \in E$. Otherwise, we have $A_G[i, j] = 0$. Particularly, the diagonal entries are defined to be $A_G[i, i] = 0$ since we do not allow self-loops.

For two vertices $u, v \in V$, the distance between u and v in G is defined as the length of a shortest path between u and v in G. This length is denoted by $d_G(u, v)$. We define the distance matrix $D_G \in \mathbb{N}^{n \times n}$ by $D_G[i, j] = d_G(v_i, v_j)$. Obviously, D_G is a symmetric matrix with non-negative entries. Furthermore, for any given spanning tree T of G, it holds $D_T[i, j] \geq D_G[i, j]$ for all $v_i, v_j \in V$.

Let $A \in \mathbb{N}^{n \times m}$ and $B \in \mathbb{N}^{n \times m}$ be two $n \times m$-matrices. We denote by $C = A \circ B$ the matrix we obtain by performing a multiplication element by element, i.e., we have $C[i, j] = A[i, j] \cdot B[i, j]$ for all pairs (i, j). This entrywise product of two matrices of equal dimensions is also known as the Hadamard or Schur product. For evaluating a matrix $A \in \mathbb{N}^{n \times n}$, we use the L_p-norm ($1 \leq p < \infty$), which is defined as

$$\|A\|_{L_p} = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} A[i, j]^p \right)^{1/p} .$$

The problem of computing a graph-approximating spanning tree is formally characterized as follows.

Problem:	GAST (with respect to $\|\cdot\|_{L_p}$).
Input:	A connected graph G and an algebraic number γ.
Question:	Is there a spanning tree T of G with $\|(D_T - D_G) \circ A_G\|_{L_p} \leq \gamma$?

Now, we show that computing such a tree is hard under the L_p-norm for all $p \in \mathbb{N}$.

Theorem 6.2. GAST *with respect to* $\|\cdot\|_{L_p}$ *is* $\mathcal{NP}$*-complete for all* $p \in \mathbb{N}$.

Proof. The containment in $\mathcal{NP}$ is obvious. We prove the hardness by reduction from 2HS using the graph representation $G(\mathcal{C}, \mathcal{S})$. The idea of the reduction is simple: We join the end vertices of $G(\mathcal{C}, \mathcal{S})$, v_0' and v_n', by a connection consisting of a couple of paths. Moreover, the same technique is used to force the clause path edges of $G(\mathcal{C}, \mathcal{S})$ into any optimal spanning tree. The number of additional paths and their length depends on the given 2HS instance and is polynomial in the number of literals in $\mathcal{S}$ and clauses in $\mathcal{C}$.

Let $(\mathcal{C}, \mathcal{S}, k)$ be an instance of 2HS and let N be the number of vertices in its graph representation $G(\mathcal{C}, \mathcal{S})$. We define the graph $G = (V, E)$ such that V consists of the vertices in the graph $G(\mathcal{C}, \mathcal{S})$ and

- N_A vertex sets $\{a_{\mu,1}, a_{\mu,2}, \ldots, a_{\mu,L_A}\}$ for each $\mu \in \{1, \ldots, N_A\}$ and
- $N_B \cdot K$ vertex sets $\{b^\mu_{\nu,1}, b^\mu_{\nu,2}, \ldots, b^\mu_{\nu,L_B}\}$ for each $\mu \in \{1, \ldots, K\}$ and each $\nu \in \{1, \ldots, N_B\}$

where K is the number of clause path edges in $G(\mathcal{C}, \mathcal{S})$ and $N_A, N_B, L_A, L_B \in \mathbb{N}$ are four parameters which will be chosen later. The edge set E consists of all edges in $G(\mathcal{C}, \mathcal{S})$ and

- N_A paths $A_\mu = (v_0', a_{\mu,1}, a_{\mu,2}, \ldots, a_{\mu,L_A}, v_n')$ for each $\mu \in \{1, \ldots, N_A\}$ and
- $N_B \cdot K$ paths $B^\mu_\nu = (u, b^\mu_{\nu,1}, b^\mu_{\nu,2}, \ldots, b^\mu_{\nu,L_B}, v)$ for each $\mu \in \{1, \ldots, K\}$ and for each $\nu \in \{1, \ldots, N_B\}$ where $\{u, v\}$ is a clause path edge.

The gadget is illustrated in Figures 6.2 and 6.3 where the latter figure is a detailed view onto the extension of a clause path edge which is not displayed in Figure 6.2. Obviously, the number of vertices and edges in G are polynomial in n and m if N_A, N_B, L_A, and L_B are polynomial in n and m.

Now, we define

$$\gamma = N^{p+1} + N_A \cdot (L_A + n(m+2) + k)^p + K \cdot N_B \cdot L_B^p$$

and we claim that G has a spanning tree T such that $\|(D_T - D_G) \circ A_G\|^p_{L_p} \leq \gamma$ if and only if $(\mathcal{C}, \mathcal{S}, k)$ has an admissible 2HS-solution $\mathcal{S}'$ of size $|\mathcal{S}'| \leq k$.

Claim 6.3. *Let* $(\mathcal{C}, \mathcal{S}, k)$ *be an instance of* 2HS. *Then,* $G = (V, E)$ *has a spanning tree* T *such that*

$$\|(D_T - D_G) \circ A_G\|^p_{L_p} \leq \gamma$$

if $(\mathcal{C}, \mathcal{S}, k)$ *has an admissible solution* $\mathcal{S}'$ *of size* $|\mathcal{S}'| \leq k$.

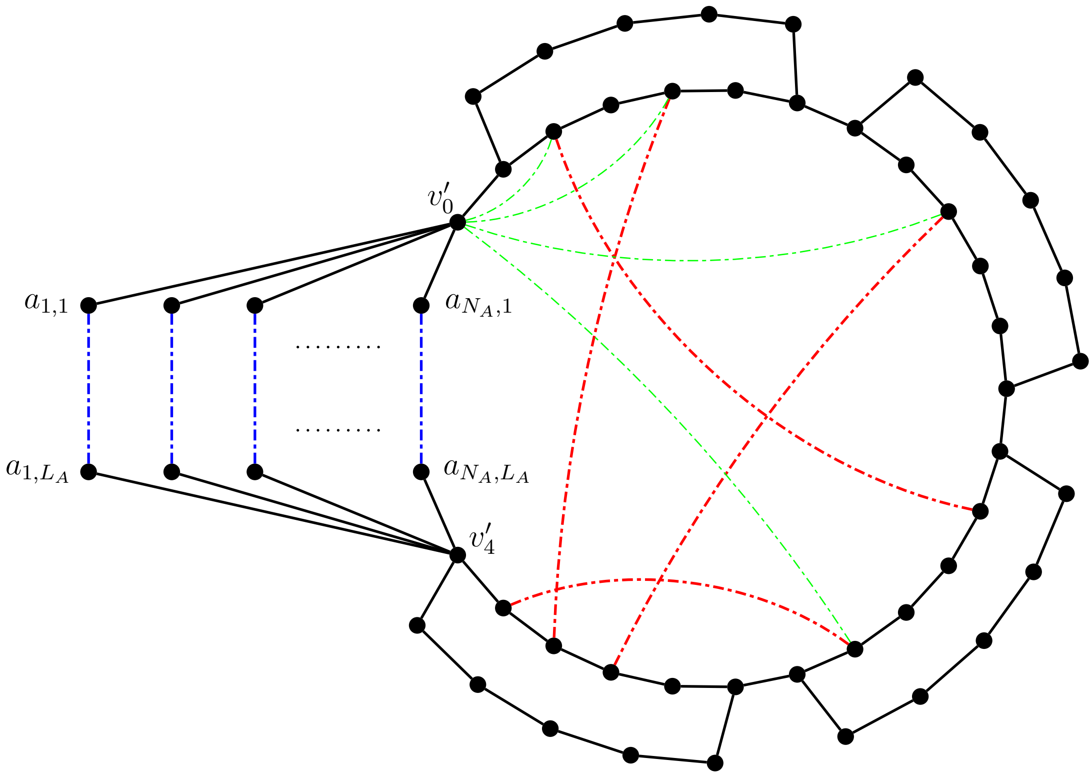


Figure 6.2: Extended graph representation of a 2HS instance.

Proof. Let $\mathcal{S}'$ be an admissible solution of $(\mathcal{C}, \mathcal{S}, k)$ such that $|\mathcal{S}'| \leq k$. We construct the spanning tree T of G as follows: For the part of G which corresponds to $G(\mathcal{C}, \mathcal{S})$, we use the spanning tree $T_{G(\mathcal{C},\mathcal{S})}$ according to Lemma 6.1. In that construction, we remove two edges for each clause in $\mathcal{C}$ as well as one edge for each literal in $\mathcal{S}$ of the given 2HS instance $(\mathcal{C}, \mathcal{S}, k)$. Thus, if N is the number of vertices in $G(\mathcal{C}, \mathcal{S})$, it holds that

$$\sum_{\{u,v\} \in G(\mathcal{C},\mathcal{S})} (d_T(u,v) - 1)^p \leq N \cdot N^p = N^{p+1} \,. \tag{6.1}$$

Note that $\{u, v\} \in G(\mathcal{C}, \mathcal{S})$ denotes all edges in $G(\mathcal{C}, \mathcal{S})$ (it does *not* mean all pairs of vertices). For the sake of readability, we assume this meaning of the notation unless stated otherwise.

Additionally, we break the paths A_μ and B_κ^ν as follows:

1. For each path A_μ, $\mu \in \{1, \ldots N_A\}$, we remove the edge $\{a_{\mu,1}, a_{\mu,2}\}$.

2. For each path B_κ^ν, $\nu \in \{1, \ldots, K\}$ and $\kappa \in \{1, \ldots, N_B\}$, we remove the edge $\{b_{\kappa,1}^\nu, b_{\kappa,2}^\nu\}$.

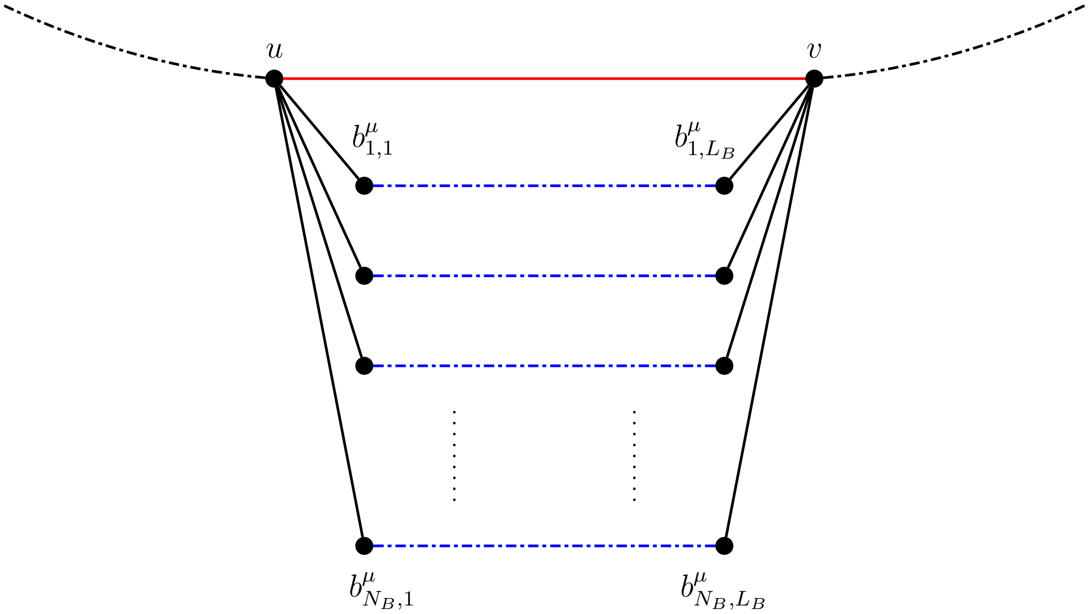


Figure 6.3: Extension paths of a clause path edge $\{u, v\}$.

For the first construction rule, the contribution to $\|(D_T - D_G) \circ A_G\|^p_{L_p}$ is bounded by

$$\sum_{\mu=1}^{N_A} \sum_{\{u,v\} \in A_\mu} (d_T(u,v) - 1)^p \leq N_A \cdot (L_A + n(m+2) + k)^p \tag{6.2}$$

and for the second rule, we obtain

$$\sum_{\nu=1}^{K} \sum_{\kappa=1}^{N_B} \sum_{\{u,v\} \in B^{\nu}_{\kappa}} (d_T(u,v) - 1)^p \leq K \cdot N_B \cdot L^p_B \; . \tag{6.3}$$

Combining (6.1), (6.2), and (6.3), we get the required quality of T

$$\sum_{\{u,v\} \in E} (d_T(u,v) - 1)^p \leq N^{p+1} + N_A \cdot (L_A + n(m+2) + k)^p + K \cdot N_B \cdot L^p_B = \gamma \; .$$

□

Claim 6.4. *Let $(\mathcal{C}, \mathcal{S}, k)$ be an instance of* 2HS*. Then, $(\mathcal{C}, \mathcal{S}, k)$ has an admissible solution $\mathcal{S}'$ of size $|\mathcal{S}'| \leq k$ if the graph G has a spanning tree T such that*

$$\|(D_T - D_G) \circ A_G\|^p_{L_p} \leq \gamma$$

with $N_A > N^{p+1}$, $N_B > N_A \cdot (2N^2)^p$, $L_A = N^2 \cdot (n(m+2) + k)$, and $L_B > N^2 + 1$.

Proof. Let T be a spanning tree of $G = (V, E)$ such that

$$\|(D_T - D_G) \circ A_G\|^p_{L_p} \leq \gamma = N^{p+1} + N_A \cdot (L_A + n(m+2) + k)^p + K \cdot N_B \cdot L^p_B \; . \tag{6.4}$$

Assume, there is a clause path edge which does not belong to T. We consider the extension paths of this edge (see Figure 6.3) and distinguish two different cases. Either there is exactly one path B_κ^ν ($\kappa \in \{1, \ldots, N_B\}$ and ν depends on the removed clause path edge), which is intact, or each of these paths is broken. Note that two or more intact extension paths of the same clause path edge would imply the existence of a cycle. For the first case, we can lower bound the contribution to $\|(D_G - D_T) \circ A_G\|_{L_p}^p$ for removing an edge from A_μ, $\mu \in \{1, \ldots, N_A\}$, by

$$\sum_{\mu=1}^{N_A} \sum_{\{u,v\} \in A_\mu} (d_T(u,v) - 1)^p \geq N_A \cdot (L_A + n(m+2))^p \tag{6.5}$$

and for all deleted edges of paths B_κ^ν, $\nu \in \{1, \ldots, K\}$ and $\kappa \in \{1, \ldots, N_B\}$, we obtain

$$\begin{aligned} \sum_{\nu=1}^{K} \sum_{\kappa=1}^{N_B} \sum_{\{u,v\} \in B_\kappa^\nu} (d_T(u,v) - 1)^p &\geq (K-1) \cdot N_B \cdot L_B^p + (N_B - 1) \cdot (2L_B)^p + L_B^p \\ &= K \cdot N_B \cdot L_B^p + (2^p - 1) \cdot (N_B - 1) \cdot L_B^p . \end{aligned} \tag{6.6}$$

By assumption, we have $N_A > N^{p+1}$, $L_B > N^2 + 1$, $N_B > N_A \cdot (2N^2)^p$ with $p \in \mathbb{N}$. Thus,

$$N_A \cdot (N^2 \cdot (n(m+2) + k) + n(m+2))^p > N^{p+1}$$

and

$$(2^p - 1) \cdot (N_B - 1) \cdot L_B^p > N_A \cdot ((N^2 + 1) + (n(m+2) + k))^p$$

imply the following contradiction to (6.4):

$$\begin{aligned} & N_A \cdot (N^2 \cdot (n(m+2) + k) + n(m+2))^p + (2^p - 1) \cdot (N_B - 1) \cdot L_B^p \\ & \qquad > N^{p+1} + N_A \cdot ((N^2 + 1) \cdot (n(m+2) + k))^p \\ \Longrightarrow \quad & N_A \cdot (L_A + n(m+2))^p + K \cdot N_B \cdot L_B^p + (2^p - 1) \cdot (N_B - 1) \cdot L_B^p \\ & \qquad > N^{p+1} + N_A \cdot (L_A + n(m+2) + k)^p + K \cdot N_B \cdot L_B \\ \Longrightarrow \quad & \|(D_T - D_G) \circ A_G\|_{L_p}^p > \gamma . \end{aligned}$$

For the second case where each path $B_\kappa^{\{u,v\}}$ with $\kappa \in \{1, \ldots, N_B\}$ is broken, the lower bound in (6.5) holds and (6.6) changes to

$$\begin{aligned} \sum_{\nu=1}^{K} \sum_{\kappa=1}^{N_B} \sum_{\{u,v\} \in B_\kappa^\nu} (d_T(u,v) - 1)^p &\geq (K-1) \cdot N_B \cdot L_B + N_B \cdot (L_B + 2n(m+2))^p \\ &= K \cdot N_B \cdot L_B^p + N_B \cdot (2n(m+2))^p . \end{aligned}$$

Analogously to the previous case, we get a contradiction to (6.4). Thus, all K clause path edges are forced into each optimal spanning tree by the extension paths B_ν^μ, $\mu \in \{1, \ldots, K\}$ and $\nu \in \{1, \ldots, N_B\}$.

Now, we turn our attention to the paths A_μ, $\mu \in \{1, \ldots, N_A\}$, and show that all of these paths must be broken. Afterwards, we prove that the distance between v_0' and v_n'

is at most $n(m+2)+k$. Since all clause path edges belong to T, we are able to apply Lemma 6.1 in order to prove the existence of an admissible solution $\mathcal{S}'$ of size $|\mathcal{S}'| \le k$ for the 2HS instance $(\mathcal{C}, \mathcal{S}, k)$. To this end, we first assume that there is an intact path A_μ for some $\mu \in \{1, \ldots, N_A\}$. The contribution of all broken paths A_μ includes the length of the intact path:

$$\sum_{\mu=1}^{N_A} \sum_{\{u,v\} \in A_\mu} (d_T(u,v) - 1)^p \ge (N_A - 1) \cdot (2L_A)^p \,.$$

Since the direct connection between v_0' and v_n' (inside the original 2HS gadget) is broken, there exists some edge $\{u, v\} \in G(\mathcal{C}, \mathcal{S})$ such that

$$(d_T(u,v) - 1)^p \ge L_A^p \,.$$

This is a contradiction to (6.4) since

$$\begin{aligned}
& \|(D_T - D_G) \circ A_G\|_{L_p}^p > \gamma \\
\Longleftarrow\quad & (N_A - 1) \cdot (2L_A)^p + L_A^p + K \cdot N_B \cdot L_B^p \\
& \qquad\qquad > N \cdot N^{p+1} + N_A \cdot (L_A + n(m+2) + k)^p + K \cdot N_B \cdot L_B^p \\
\Longleftarrow\quad & (N_A - 1) \cdot (2N^2 \cdot (n(m+2)+k))^p + (N^2 \cdot (n(m+2)+k))^p \\
& \qquad\qquad > N^{p+1} + N_A \cdot ((N^2+1) \cdot (n(m+2)+k))^p \\
\Longleftarrow\quad & (N_A - 1) \cdot 2^p \cdot N^{2p} > N_A \cdot (N^2+1)^p \\
\Longleftarrow\quad & N^2 \cdot 2^p \cdot N^{2p} > (N^2+1)^{p+1} \\
\Longleftarrow\quad & N^2 \cdot 2^{\frac{p}{p+1}} > N^2 + 1
\end{aligned}$$

which is true if $N_A > N^{p+1}$, $L_A = N^2 \cdot (n(m+2)+k)$, $N > 1$, and $p \in \mathbb{N}$.

Closing the proof, we assume that the distance between v_0' and v_n' is greater than $n(m+2)+k$. Choosing $N_A > N^{p+1}$, we obtain a contradiction to (6.4) since

$$N_A \cdot (L_A + n(m+2) + k + 1)^p > N^{p+1} + N_A \cdot (L_A + n(m+2) + k)^p$$

implies $\|(D_T - D_G) \circ A_G\|_{L_p}^p > \gamma$. □

This proves the theorem. □

6.4 Approximating GAST with respect to L_1

The problem GAST is related to the problem of computing a so-called fundamental cycle basis of minimum weight. Let $G = (V, E)$ be an undirected graph and T be a spanning tree of G. There are $m - n + 1$ edges $e_1, \ldots, e_{m-n+1}$ in E which do not belong to T. When adding the edge e_i to T, we obtain a cycle $c_i = C(T, e_i)$ for all $i = 1, \ldots, m-n+1$. The set $\mathcal{C} = \{c_1, \ldots, c_{m-n+1}\}$ is called a *fundamental cycle basis* of G with respect to T. Fundamental cycle bases were studied by Sysło [Sys79; Sys82].

Given a weighted undirected graph $G = (V, E, w)$, the weight of a fundamental cycle basis $\mathcal{C}$ is defined to be

$$fund_G(\mathcal{C}) = \sum_{i=1}^{m-n+1} w(c_i) \tag{6.7}$$

where $w(c_i)$ denotes the sums of all edge weights in the cycle c_i. Since a fundamental cycle basis can be defined by a spanning tree, we also use the notation $fund_G(T)$. Furthermore, we omit the index 'G' of the graph if it is clear from the context. Using this notation, the minimum fundamental cycle basis problem (MIN-FCB) is defined as:

Problem: MIN-FCB.
Input: A connected and weighted graph $G = (V, E, w)$.
Output: A fundamental cycle basis $\mathcal{C} = \{c_1, \ldots, c_{m-n+1}\}$ of minimum weight.

The performance of a polynomial-time approximation algorithm $\mathcal{A}$ is measured by its *approximation ratio* $\rho(n)$. We say that $\mathcal{A}$ approximates some optimization problem within $\rho(n)$ if for any input of size n, the cost c of the solution computed by $\mathcal{A}$ is within a factor of $\rho(n)$ of the cost c^* of an optimal solution, i.e., we have

$$\max\left(\frac{c}{c^*}, \frac{c^*}{c}\right) \leq \rho(n) \ .$$

Now, we show the approximability of GAST by a reduction to MIN-FCB.

Theorem 6.5. *If* MIN-FCB *is approximable within* $\rho > 1$ *then* GAST *with respect to* $\|\cdot\|_{L_1}$ *is approximable within* 3ρ.

Proof. Let $G = (V, E)$ be an instance of GAST with respect to L_1-norm, that is we want to compute a spanning tree T of G such that $\|(D_T - D_G) \circ A_G\|_{L_1}$ is minimized. Without loss of generality, we assume G is a simple and 2-edge-connected graph. Otherwise, each bridge e of G is contained in every spanning tree T, thus, we can consider the graph G/e in which the edge e is contracted. We show that an approximation algorithm for the minimum fundamental cycle basis problem with ratio $\rho > 1$ can be used to construct an algorithm for GAST with approximation ratio of at most 3ρ. To this end, we transform the instance $G = (V, E)$ for GAST into an instance $G' = (V', E', w)$ for MIN-FCB. The graph G remains unchanged, thus, we have $V' = V$ and $E' = E$. We introduce a weight function $w\colon E \to \mathbb{R}^+$ with $w(e) = 1$ for all $e \in E$.

For any spanning tree T of G (or G', respectively), we denote by $fund_{G'}(T)$ and $gast_G(T)$ the objective function with respect to MIN-FCB and GAST, i.e., $fund_{G'}(T)$ is equivalent to (6.7) and it holds that

$$gast_G(T) = \|(D_T - D_G) \circ A_G\|_{L_1} = \sum_{\{u,v\} \in E} (d_T(u, v) - 1) \ .$$

We observe that each spanning tree T satisfies

$$fund_{G'}(T) = gast_G(T) + 2 \cdot w(E \setminus T)$$

and

$$w(E \setminus T) \leq gast_G(T) \ .$$

Let T^*_{GAST} be a spanning tree in G minimizing the function $gast_G(T)$ and let $T^*_{\text{Min-FCB}}$ be a spanning tree in G' minimizing the function $fund_{G'}(T)$, respectively. Suppose MIN-FCB is approximable within $\rho > 1$, that is, there exists a polynomial-time algorithm computing a spanning tree $T^{'}_{\text{Min-FCB}}$ of G' such that

$$fund_{G'}(T^{'}_{\text{Min-FCB}}) \leq \rho \cdot fund_{G'}(T^*_{\text{Min-FCB}}) \ .$$

Then, it holds that

$$\begin{aligned}
gast_G(T^{'}_{\text{Min-FCB}}) &= fund_{G'}(T^{'}_{\text{Min-FCB}}) - w(E \setminus T^{'}_{\text{Min-FCB}}) \\
&\leq \rho \cdot fund_{G'}(T^*_{\text{Min-FCB}}) - w(E \setminus T^{'}_{\text{Min-FCB}}) \\
&\leq \rho \cdot fund_{G'}(T^*_{\text{GAST}}) - w(E \setminus T^{'}_{\text{Min-FCB}}) \\
&= \rho \cdot (gast_G(T^*_{\text{GAST}}) + 2 \cdot w(E \setminus T^*_{\text{GAST}})) - w(E \setminus T^{'}_{\text{Min-FCB}}) \\
&= \rho \cdot gast_G(T^*_{\text{GAST}}) + (2\rho - 1) \cdot w(E \setminus T^*_{\text{GAST}}) \\
&\leq 3\rho \cdot gast_G(T^*_{\text{GAST}}) \ .
\end{aligned}$$

This proves the theorem. □

Galbiati and Amaldi [GA04] showed that the problem MIN-FCB can be approximated within $2^{\mathcal{O}(\sqrt{\log n \log \log n})}$ for arbitrary weighted graphs. This result was obtained by a reduction to the problem of computing a minimum communication cost spanning tree (MCT) [Res99]. The currently best known approximation algorithm is due to Elkin, Emek, Spielman, and Teng [EEST08] achieving an approximation ratio of $\mathcal{O}(\log^2 n \log \log n)$. Hence, we obtain the following corollary.

Corollary 6.6. GAST *with respect to* $\|\cdot\|_{L_1}$ *is approximable within* $\mathcal{O}(\log^2 n \log \log n)$.

Chapter 7

Conclusion

In this thesis, we analyzed spanning trees of weighted bispanning graphs in order to make progress in proving a conjecture by Mayr and Plaxton [MP92]. Another formulation of this conjecture states that each weighted bispanning graph $B = (V, P, Q)$ which satisfies $w(P) < w(Q)$ and Q is the only spanning tree with weight $w(Q)$, has at least $|V| - 1$ spanning trees with pairwise different weights. We were able to prove this conjecture if the spanning tree P also has unique weight or if the cycle matroid of B is strongly base orderable. As a consequence, it is now sufficient to analyze bispanning graphs which contain a minor isomorphic to the complete graph on four vertices K_4.

Based on these findings, we refined Mayr and Plaxton's conjecture, that is, we now conjecture that it is sufficient to count only spanning trees which define a new partition of a bispanning graph. Here, we formulated a decomposition theorem and proved that we only have to analyze so-called atomic bispanning graphs, which are a subset of all bispanning graphs that have a minor isomorphic to K_4 (the smallest non-trivial atomic bispanning graph). In particular, we showed that any partition of K_4 into P and Q, together with any weight function achieving the above requirements, also satisfies the slightly stronger conjecture.

Furthermore, we considered base orderings, which are related to paths in the tree graph. We discovered a new property, which we called subsequence-interchangeable, in order to further classify so-called cyclic base orderings. Such a subsequence-interchangeable base ordering corresponds to a path between two spanning trees T and T' in the tree graph which has the property that each connected sub-path (actually its corresponding edge swaps) defines another path with respect to T. We analyzed the operations 'double-leaf attachment' and 'edge-split' in order to show how to construct these base orderings. Again, we pointed out that it suffices only to consider atomic bispanning graphs like in the previous problem of counting weighted spanning trees. Actually, there seems to be a stronger connection between these two problems since all spanning trees lying on a path which corresponds to a subsequence-interchangeable base ordering have pairwise different weights. Hence, it might be possible to use these subsequence-interchangeable base orderings in order to find a proof of Mayr and Plaxton's conjecture.

Finally, we considered a network sparsification problem: compute a spanning tree T of a given undirected graph $G = (V, E)$ minimizing the sum of distances between all pairs of vertices which are connected by an edge in the original graph. Here, we proved that this problem is $\mathcal{NP}$-complete with respect to the L_p-norm for all $p \in \mathbb{N}$ by using a clever extension of the results in [EKM$^+$08]. For $p = 1$, we showed a reduction to the problem of computing a minimum fundamental cycle basis implying an approximation ratio of $\mathcal{O}(\log^2 n \log \log n)$.

Bibliography

[Bau08] Matthias Baumgart. Partitioning bispanning graphs into spanning trees. In *Proceedings of the 19th International Workshop on Combinatorial Algorithms (IWOCA'08)*, pages 50–62, 2008.

[BD72] Richard A. Brualdi and George W. Dinolt. Characterization of transversal matroids and their presentations. *Journal of Combinatorial Theory (B)*, 12:268–286, 1972.

[BM97] Andrei Z. Broder and Ernst W. Mayr. Counting minimum weight spanning trees. *Journal of Algorithms*, 24(1):171–176, 1997.

[Bon72] John A. Bondy. Transversal matroids, base-orderable matroids, and graphs. *Quarterly Journal of Mathematics Oxford*, 23(1):81–89, 1972.

[Bor26] Otakar Borůvka. O jistém problému minimálním. *Práca Moravské Přírodovědecké Společnosti*, 3:37–58, 1926. (in Czech).

[Bru69] Richard A. Brualdi. Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society*, 1:161–167, 1969.

[Bru71] Richard A. Brualdi. Induced matroids. *Proceedings of the American Mathematical Society*, 29(2):213–221, 1971.

[Bru74] Richard A. Brualdi. Networks and the Shannon switching game. *Delta (Waukesha)*, 4:1–23, 1974.

[Bru77] Richard A. Brualdi. *Introductory Combinatorics*. North-Holland, New York, NY, 1977.

[Bry73] Thomas H. Brylawski. Some properties of basic families of subsets. *Discrete Mathematics*, 6:333–341, 1973.

[BS68] Richard A. Brualdi and Edward B. Scrimger. Exchange systems, matchings, and transversals. *Journal of Combinatorial Theory*, 5:244–257, 1968.

[BT08] Matthias Baumgart and Hanjo Täubig. The complexity of computing graph-approximating spanning trees. Technical Report TUM-I0822, Technische Universität München, Institut für Informatik, 2008.

[BW70] John Bruno and Louis Weinberg. A constructive graph-theoretic solution of the Shannon switching game. *IEEE Transactions on Circuit and Systems*, 17(1):74–81, 1970.

[BW71a] John A. Bondy and Dominic J. A. Welsh. Some results on transversal matroids and constructions for identically self-dual matroids. *Quarterly Journal of Mathematics Oxford*, 22(3):435–451, 1971.

[BW71b] John Bruno and Louis Weinberg. The principal minors of a matroid. *Linear Algebra and its Applications*, 4(1):17–54, 1971.

[CC96] Sun Chung and Anne Condon. Parallel implementation of Borůvka's minimum spanning tree algorithm. In *Proceedings of the 10th International Parallel Processing Symposium (IPPS'96)*, pages 302–308, 1996.

[CGM80] Paolo M. Camerini, Giulia Galbiati, and Francesco Maffioli. Complexity of spanning tree problems: Part I. *European Journal of Operational Research*, 5(5):346–352, 1980.

[CH80] Jens Clausen and Lone A. Hansen. Finding k edge-disjoint spanning trees of minimum total weight in a network: An application of matroid theory. In *Proceedings of the 2nd British Conference on Combinatorial Optimization (CO'79)*, volume 13 of *Mathematical Programming Study*, pages 88–101, 1980.

[Cha72] Stephen M. Chase. An implemented graph algorithm for winning Shannon switching games. *Communications of the ACM*, 15(4):253–256, 1972.

[Cha00] Bernard Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *Journal of the ACM*, 47(6):1028–1047, 2000.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, second edition, 2001.

[DDGS03] Elias Dahlhaus, Peter Dankelmann, Wayne Goddard, and Henda C. Swart. MAD trees and distance-hereditary graphs. *Discrete Applied Mathematics*, 131(1):151–167, 2003.

[Deo74] Narsingh Deo. *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall, Englewood Cliffs, NJ, 1974.

[Die05] Reinhard Diestel. *Graph Theory*. Springer, Berlin, Heidelberg, third edition, 2005.

[Din70] Yefim Dinitz. Algorithm for solution of a problem of maximum flow in a network with power estimation. *Soviet Mathematics Doklady*, 11(5):1277–1280, 1970.

[Dir52] Gabriel A. Dirac. A property of 4-chromatic graphs and some remarks on critical graphs. *Journal of the London Mathematical Society*, 27:85–92, 1952.

[DPK82] Narsingh Deo, Gurpur Prabhu, and Mukkai S. Krishnamoorthy. Algorithms for generating fundamental cycles in a graph. *ACM Transactions on Mathematical Software*, 8(1):26–42, 1982.

[DRT92] Brandon Dixon, Monika Rauch, and Robert E. Tarjan. Verification and sensitivity analysis of minimum spanning trees in linear time. *SIAM Journal on Computing*, 21(6):1184–1192, 1992.

[Edm65a] Jack Edmonds. Lehman's switching game and a theorem of Tutte and Nash-Williams. *Journal of Research of the National Bureau of Standards*, 69B:73–77, 1965.

[Edm65b] Jack Edmonds. Minimum partition of a matroid into independent subsets. *Journal of Research of the National Bureau of Standards*, 69B:67–72, 1965.

[EEST08] Michael Elkin, Yuval Emek, Daniel A. Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. *SIAM Journal on Computing*, 38(2):608–628, 2008.

[EF65] Jack Edmonds and Delbert R. Fulkerson. Transversal and matroid partition. *Journal of Research of the National Bureau of Standards*, 69B:147–153, 1965.

[EKM+08] Stefan Eckhardt, Sven Kosub, Moritz G. Maaß, Hanjo Täubig, and Sebastian Wernicke. Combinatorial network abstraction by trees and distances. *Theoretical Computer Science*, 407(1-3):1–20, 2008.

[Elk06] Michael Elkin. A faster distributed protocol for constructing a minimum spanning tree. *Journal of Computer and System Sciences*, 72(8):1282–1308, 2006.

[Eve79] Shimon Even. *Graph Algorithms*. Computer Science Press, Rockville, MD, 1979.

[FF62] Lester R. Ford and Delbert R. Fulkerson. *Flows and Networks*. Princeton University Press, Princeton, NJ, 1962.

[FRS85] Martin Farber, Bruce Richter, and Herbert Shank. Edge-disjoint spanning trees: A connectedness theorem. *Journal of Graph Theory*, 9(3):319–324, 1985.

[GA04] Giulia Galbiati and Edoardo Amaldi. On the approximability of the minimum fundamental cycle basis problem. In *Proceedings of the 1st International Workshop on Approximation and Online Algorithms (WAOA'03)*, volume 2909 of *Lecture Notes in Computer Science*, pages 151–164. Springer, 2004.

[Gab76] Harold N. Gabow. Decomposing symmetric exchanges in matroid bases. *Mathematical Programming*, 10:271–276, 1976.

[Gab91] Harold N. Gabow. A matroid approach to finding edge connectivity and packing arborescences. In *Proceedings of the 23rdAnnual ACM Symposium on Theory of Computing (STOC'91)*, pages 112–122. ACM, 1991.

[Gam99] Anna Gambin. On approximating the number of bases of exchange preserving matroids. In *Proceedings of the 24th International Symposium on Mathematical Foundations of Computer Science (MFCS'99)*, volume 1672 of *Lecture Notes in Computer Science*, pages 332–342. Springer, 1999.

[GH85] Ronald L. Graham and Pavol Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1):43–57, 1985.

[GHS83] Robert G. Gallager, Pierre A. Humblet, and Philip M. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 5(1):66–77, 1983.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractabilit: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, 1979.

[Gre73] Curtis Greene. A multiple exchange property for bases. *Proceedings of the American Mathematical Society*, 39(1):45–50, 1973.

[GS85] Harold N. Gabow and Matthias Stallmann. Efficient algorithms for graphic matroid intersection and parity. In *Proceedings of the 12th International Colloquium on Automata Languages and Programming (ICALP'85)*, volume 194 of *Lecture Notes in Computer Science*, pages 210–220. Springer, 1985.

[Gus83] Dan Gusfield. Connectivity and edge-disjoint spanning trees. *Information Processing Letters*, 16(2):87–89, 1983.

[GW92] Harold N. Gabow and Herbert H. Westermann. Forests, frames, and games: Algorithms for matroid sums and applications. *Algorithmica*, 7(5&6):465–497, 1992.

[HK73] John Hopcroft and Richard Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[HT95] Refael Hassin and Arie Tamir. On the minimum diameter spanning tree problem. *Information Processing Letters*, 53(2):109–111, 1995.

[Hu74] Te C. Hu. Optimum communication spanning trees. *SIAM Journal on Computing*, 3(3):188–195, 1974.

[Ima83] Hiroshi Imai. Network-flow algorithms for lower-truncated transversal polymatroids. *Journal of the Operations Research Society of Japan*, 26(3):186–211, 1983.

[Ing75] Aubrey W. Ingleton. Non-base-orderable matroids. In *Proceedings of the 5th British Combinatorial Conference (BCC'75)*, volume 15 of *Congressus Numerantium*, pages 355–359, 1975.

[Jar30] Vojtěch Jarník. O jistém problému minimálním. *Práca Moravské Přírodovědecké Společnosti*, 6:57–63, 1930. (in Czech).

[JLRK78] David S. Johnson, Jan K. Lenstra, and Alexander H. G. Rinnooy Kan. The complexity of the network design problem. *Networks*, 8(4):279–285, 1978.

[JM95] Donald B. Johnson and Panagiotis Metaxas. A parallel algorithm for computing minimum spanning trees. *Journal of Algorithms*, 19(3):383–401, 1995.

[Kan87] Mikio Kano. Maximum and k-th maximal spanning trees of a weighted graph. *Combinatorica*, 7(2):205–214, 1987.

[Kin97] Valerie King. A simpler minimum spanning tree verification algorithm. *Algorithmica*, 18(2):263–270, 1997.

[KK67] Genya Kishi and Yoji Kajitani. On maximally distinct trees. In *Proceedings of the 5th Annual Allerton Conference on Circuit and System Theory*, pages 635–643, 1967.

[KK68] Genya Kishi and Yoji Kajitani. Maximally distinct trees in a linear graph. *Electronics and Communications in Japan*, 51A(5):35–42, 1968. (in Japanese).

[KK69] Genya Kishi and Yoji Kajitani. Maximally distant trees and principal partition of a linear graph. *IEEE Transactions on Circuit and Systems*, 16(3):323–330, 1969.

[KKS78] Tatsuya Kawamoto, Yoji Kajitani, and Shoji Shinoda. On the second maximal spanning trees of a weighted graph. *Transactions of the IECE of Japan*, 61-A:988–995, 1978. (in Japanese).

[KKT95] David R. Karger, Philip N. Klein, and Robert E. Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM*, 42(2):321–328, 1995.

[Kom85] Janos Komlós. Linear verification for spanning trees. *Combinatorica*, 5(1):57–65, 1985.

[KPS00] Judith Keijsper, Rudi Pendavingh, and Alexander Schrijver. Adjacency, inseparability, and base orderability in matroids. *European Journal of Combinatorics*, 21(4):487–502, 2000.

[Kru56] Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

[KT73] Tiko Kameda and Shunichi Toida. Efficient algorithms for determining an extremal tree of a graph. In *Proceedings of the 14th Annual Symposium on Foundations of Computer Science (FOCS'73)*, pages 12–15, 1973.

[KUM88] Yoji Kajitani, Shuichi Ueno, and Hiroshi Miyano. Ordering of the elements of a matroid such that its consecutive w elements are independent. *Discrete Mathematics*, 72:187–194, 1988.

[Law76] Eugene L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart, and Winston, New York, NY, 1976.

[Leh64] Alfred Lehman. A solution of the Shannon switching game. *SIAM Journal on Applied Mathematics*, 12(4):687–725, 1964.

[LM82] Eugene L. Lawler and Charles U. Martel. Flow network formulations of polymatroid optimization problems. In *Bonn Workshop on Combinatorial Optimization*, volume 16 of *Annals of Discrete Mathematics*, pages 189–200, 1982.

[LPSPP05] Zvi Lotker, Boaz Patt-Shamir, Elan Pavlov, and David Peleg. Minimum-weight spanning tree construction in $\mathcal{O}(\log \log n)$ communication rounds. *SIAM Journal on Computing*, 35(1):120–131, 2005.

[Man96] Richard Mansfield. Strategies for the Shannon switching game. *The American Mathematical Monthly*, 103(3):250–252, 1996.

[Min81] Edward Minieka. A polynomial time algorithm for finding the absolute center of a network. *Networks*, 11(4):351–355, 1981.

[Mir71] Leon Mirsky. *Transversal Theory*, volume 75 of *Mathematics in Science and Engineering.* Academic Press, New York, NY, 1971.

[MP92] Ernst W. Mayr and C. Gregory Plaxton. On the spanning trees of weighted graphs. *Combinatorica*, 12(4):433–447, 1992.

[NG99] Ladislav Novak and Alan Gibbons. *Hybrid Graph Theory and Network Analysis*, volume 49 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1999.

[NW61] Crispin St. J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, 36:445–450, 1961.

[NW64] Crispin St. J. A. Nash-Williams. Decomposition of finite graphs into forests. *Journal of the London Mathematical Society*, 39:12, 1964.

[Oxl92] James G. Oxley. *Matroid Theory*. Oxford University Press, New York, NY, 1992.

[Pet99] Seth Pettie. Finding minimum spanning trees in $\mathcal{O}(m\alpha(m,n))$ time. Technical Report CS-TR-99-23, University of Texas, Austin, TX, 1999.

[PR98] David Peleg and Eilon Reshef. Deterministic polylog approximation for minimum communication spanning trees. In *Proceedings of the 25th International Colloquium on Automata Languages and Programming (ICALP'98)*, volume 1443 of *Lecture Notes in Computer Science*, pages 670–681. Springer, 1998.

[PR00] David Peleg and Vitaly Rubinovich. A near-tight lower bound on the time complexity of distributed minimum-weight spanning tree construction. *SIAM Journal on Computing*, 30(5):1427–1442, 2000.

[PR02] Seth Pettie and Vijaya Ramachandran. An optimal minimum spanning tree algorithm. *Journal of the ACM*, 49(1):16–34, 2002.

[Pri57] Robert C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36:1389–1401, 1957.

[Res99] Eilon Reshef. Approximating minimum communication cost spanning trees and related problems. Master's thesis, Weizmann Institue of Science, Rehovot, Israel, 1999.

[Ros83] James Roskind. *Application of Edge Disjoint Trees to Failure Recovery in Data Communication Networks*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1983.

[RT85] James Roskind and Robert E. Tarjan. A note on finding minimum-cost edge-disjoint spanning trees. *Mathematics of Operations Research*, 10(4):701–708, 1985.

[Sch03] Alexander Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, Berlin, Heidelberg, 2003.

[Sha55] Claude E. Shannon. Game playing machines. *Journal of the Franklin Institute*, 260(6):447–453, 1955.

[Sys79] Maciej M. Sysło. On cycle bases of a graph. *Networks*, 9(2):123–132, 1979.

[Sys82] Maciej M. Sysło. On the fundamental cycle set graph. *IEEE Transactions on Circuit and Systems*, 29(3):136–138, 1982.

[Tar72] Robert E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[Tar75] Robert E. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2):215–225, 1975.

[Tar83] Robert E. Tarjan. *Data Structures and Network Algorithms*, volume 44 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[Tut58] William T. Tutte. A homotopy theorem for matroids I and II. *Transactions of the American Mathematical Society*, 88(1):144–174, 1958.

[Tut61] William T. Tutte. On the problem of decomposing a graph into n connected factors. *Journal of the London Mathematical Society*, 36:221–230, 1961.

[Tut65] William T. Tutte. Lectures on matroids. *Journal of Research of the National Bureau of Standards*, 69B:1–48, 1965.

[Tut71] William T. Tutte. *Introduction to the Theory of Matroids*. American Elsevier, New York, NY, 1971.

[Wei97] Louis Weinberg. An optimal solution of the Shannon switching game played on a graph. In *Proceedings of the 1997 IEEE International Symposium on Circuits and Systems (ISCAS'97)*, volume 3, pages 1752–1755, 1997.

[Wel76] Dominic J. A. Welsh. *Matroid Theory*. Academic Press, London, UK, 1976.

[Wes00] Douglas B. West. *Introduction to Graph Theory*. Prentice-Hall, Upper Saddle River, NJ, 2000.

[Whi35] Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935.

[Whi80] Neil L. White. A unique exchange property for bases. *Linear Algebra and its Applications*, 31:81–91, 1980.

[Whi86] Neil L. White. *Theory of Matroids*, volume 26 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, NY, 1986.

[Wie06] Doug Wiedemann. Cyclic base orders of matroids, 2006. Available at `http://www.plumbyte.com/cyclic_base_orders_1984.pdf` (April, 2009).

[Woo74] Douglas R. Woodall. An exchange theorem for bases of matroids. *Journal of Combinatorial Theory (B)*, 16:227–228, 1974.

Index

2-HS problem, 88

adjacent, 8
admissible sequence, 60
admissible solution, 88
approximation ratio, 97
atomic, 13
augmenting swap sequence, 21

base orderable, 39
base ordering, 59
    cyclic, 60
    subsequence-interchangeable, 64
bispannable graph, 11
    atomic, 13
    composite, 13
bispanning graph, 11
    atomic, 13
    composite, 13
bridge, 9

CBO, *see* cyclic base ordering
chord, 54
clause path, 89
composite, 13
cut vertex, 9
cycle, 9
    fundamental, 9, 34
cycle matroid, 34
cyclic base ordering, 60

degree of a vertex, 9
double-leaf attachment, 14

edge
    bridge, 9
    contraction, 7
    entering, 9
    incident, 9
    leaving, 9
    subdivision, 8
edge-connectivity, 9
edge-split, 15
edge swap, 9
    admissible sequence, 60
elongation path, 88
entering edge, 9

fundamental cycle, 9, 34
fundamental cycle basis, 96

GAST problem, 91
graph
    bispannable, 11
    bispanning, 11
    connected, 9
    edge set, 7
    isomorphic, 8
    minor, 8
    simple, 7
    tree, 9
    undirected, 7
    vertex set, 7
    weighted, 7

hub, 80

incident, 9
independent set, 33
induced subgraph, 7
isomorphic, 8
isomorphism, 8

leaving edge, 9
literal gadget, 88
literal path, 88

matroid, 32
  base, 33
  cycle, 33, 34
  independent set, 33
  polygon, 34
  strongly base orderable, 36
  transversal, 35
Min-FCB problem, 97
minor, 8
  topological, 8

partition, 11
partition spanning tree, 42
path
  length, 9
  undirected, 9

safety path, 89
Shannon switching game, 23–24
SIBO, *see* subsequence-interchangeable base ordering
spanning tree
  partition, 42
spanning tree, 9
spoke, 80
strongly base orderable, 36
subdivision, 8
subgraph, 7
  induced, 7
  non-trivial, 7
subsequence-interchangeable base ordering, 64
symmetric exchange, 26
symmetric subset exchange axiom, 26

topological minor, 8
transversal matroid, 35
tree graph, 9

undirected graph, 7
undirected path, 9

vertex
  adjacent, 8
  cut, 9
  degree, 9
vertex-connectivity, 9

weighted graph, 7
weight function, 7
wheel, 80
  hub, 80
  spoke, 80