



HAL
open science

Connectivity problems in graph theory: structures, algorithms and complexity

Florian Hoersch

► **To cite this version:**

Florian Hoersch. Connectivity problems in graph theory: structures, algorithms and complexity. Computer Arithmetic. Université Grenoble Alpes [2020-..], 2021. English. NNT : 2021GRALM033 . tel-03559972

HAL Id: tel-03559972

<https://theses.hal.science/tel-03559972>

Submitted on 7 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Florian HOERSCH

Thèse dirigée par **Zoltan SZIGETI**, Professeur INP

préparée au sein du **Laboratoire des Sciences pour la Conception, l'Optimisation et la Production de Grenoble** dans l'**École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Problèmes de connexité en théorie de graphes : structures, algorithmes et complexité

connectivity problems in graph theory: structures, algorithms and complexity

Thèse soutenue publiquement le **27 septembre 2021**, devant le jury composé de :

Monsieur ZOLTAN SZIGETI

PROFESSEUR DES UNIVERSITES, GRENOBLE INP, Directeur de thèse

Monsieur TIBOR JORDAN

PROFESSEUR, Eötvös Lorand Tudományegyetem, Rapporteur

Monsieur MATTHIAS KRIESELL

PROFESSEUR, Technische Universität Ilmenau, Rapporteur

Monsieur FREDERIC HAVET

DIRECTEUR DE RECHERCHE, CNRS DELEGATION COTE D'AZUR, Examineur

Madame NADIA BRAUNER

PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES, Présidente

Monsieur STEPHAN THOMASSE

PROFESSEUR DES UNIVERSITES, ENS DE LYON, Examineur



Abstract

This thesis is concerned with 3 classes of problems related to graph connectivity.

Firstly, we deal with graph orientations where an orientation of a graph is obtained by replacing every edge by an arc between the same two vertices. This chapter is divided into two parts: one on orientations for arc-connectivity and one on orientations for vertex-connectivity. For edge-connectivity, we first review some results related to the strong orientation theorem of Nash-Williams and show that it is co-NP-complete to decide whether a given odd-vertex pairing is admissible. This resolves a question of Frank. We next show that it is NP-complete to decide whether a given graph has an orientation satisfying some arbitrary local edge-connectivity condition and give some related problems. We then give some partial results on the problem of determining whether a given graph has a strongly connected orientation such that the in-degree of every vertex is of a prescribed parity. Finally, we wish to give a connectivity property of orientations of 3-edge-connected graphs which is located between strong connectivity and 2-arc-connectivity. This problem has been suggested by Frank and leads to the introduction of a new invariant for 3-edge-connected graphs. We give several bounds for this invariant.

For vertex-connectivity, we first give an overview of previous results. We then deal with a problem suggested by Cheriyan, determining some restricted classes of Eulerian graphs all of whose Eulerian orientations are highly vertex-connected.

Next, we deal with arborescence packings. We first provide an inductive method that allows to derive theorems on packings of reachability arborescences from theorems on packings of spanning arborescences in several settings. In particular, we conclude a theorem of Kamiyama, Katoh and Takizawa from a strong form of the theorem of Edmonds. This inductive method also allows to prove a result on matroid-reachability-based packings of mixed hyperarborescences which generalizes numerous previous results. We next use matroid intersection to obtain a theorem on packing mixed hyperarborescences in a setting where the roots of the arborescences are not fixed. Finally, we provide an algorithm certifying that a problem on packing arborescences that have to satisfy a little extra condition is FPT. The same technique allows to find algorithms certifying that two similar problems are FPT. In one case, this resolves a question of Bang-Jensen, Havet and Yeo.

In the last part, we deal with connectivity augmentation problems. In particular, relying on some structure provided by Durand de Gevigney and Szigeti, we give a fast algorithm for $(2, k)$ -connectivity augmentation.

Résumé

Cette thèse traite 3 classes de problèmes liés à la connexité des graphes.

En premier lieu, nous traitons des orientations des graphes où une orientation d'un graphe non-orienté est obtenue en remplaçant chaque arête par un arc entre les mêmes deux sommets. Ce chapitre est divisé en deux parties : l'une sur les orientations pour l'arête-connexité et l'autre sur les orientations pour la sommet-connexité. Pour l'arête-connexité, nous présentons certains résultats liés au théorème fort de Nash-Williams sur les orientations et nous démontrons qu'il est co-NP-complet de décider si un "odd-vertex pairing" donné est admissible. Ceci répond à une question posée par Frank. Ensuite, nous prouvons qu'il est NP-complet de décider si un graphe a une orientation satisfaisant des conditions d'arc-connexité locale arbitraires données et nous mentionnons quelques problèmes liés. Ensuite, nous présentons certains résultats partiels sur le problème de décider si un graphe donné a une orientation fortement connexe telle que le degré entrant de chaque sommet a une certaine parité assignée. Finalement, nous souhaitons traiter une propriété de connexité des orientations des graphes 3-arête-connexes qui se situe entre la connexité forte et la 2-arc-connexité. Ce problème a été proposé par Frank et mène à l'introduction d'un nouvel invariant des graphes 3-arête-connexes. Nous démontrons plusieurs bornes pour cet invariant.

Pour la sommet-connexité, nous présentons d'abord une collection de résultats antérieurs. Puis, nous traitons un problème proposé par Cheriyan, déterminant quelques classes restreintes de graphes eulériens tel que chacune de leurs orientations eulériennes a une grande sommet-connexité.

Le chapitre suivant porte sur les packages d'arborescences. D'abord, nous proposons une méthode récursive qui permet de déduire des théorèmes sur les packages d'arborescences d'accessibilité à partir des théorèmes sur les packages d'arborescences couvrantes dans plusieurs contextes. En particulier, nous déduisons un théorème de Kamiyama, Katoh et Takizawa à partir d'une version forte du théorème d'Edmonds. Cette méthode récursive permet aussi de prouver un résultat sur les packages basés sur la matroïde-accessibilité des hyperarborescences mixtes qui généralise de nombreux résultats précédents. Puis, nous utilisons l'intersection de matroïdes afin d'obtenir un théorème sur les packages d'hyperarborescences mixtes dans un contexte où les racines des arborescences ne sont pas fixées. Finalement, nous construisons un algorithme qui montre qu'un problème sur les packages d'arborescences dans lequel les arborescences doivent satisfaire une petite condition supplémentaire est FPT. La même technique permet de montrer que deux problèmes similaires sont FPT. Dans un cas, cela répond à une question posée par Bang-Jensen, Havet et Yeo.

Dans la dernière partie, nous considérons des problèmes d'augmentation de connexité. En particulier, en nous appuyant sur des résultats structurels développés par Durand de Gevigney et Szigeti, nous donnons un algorithme rapide pour l'augmentation pour la $(2, k)$ -connexité.

Contents

1	Introduction	5
2	Preliminaries	8
2.1	Graphs	8
2.1.1	Basic notions	8
2.1.2	Operations on graphs and digraphs	12
2.1.3	Particular graphs	14
2.2	Tools	18
2.2.1	Submodularity	18
2.2.2	Splitting off	19
2.3	Complexity	20
2.3.1	Landau notation	20
2.3.2	P versus NP	20
2.3.3	A collection of NP-complete problems	21
2.3.4	XP and FPT	23
2.3.5	The Exponential Time Hypothesis (ETH)	24
2.4	Matroids	24
2.4.1	General results on matroids	25
2.4.2	Particular matroids	26
3	Orientations	32
3.1	Orientations for arc-connectivity	32
3.1.1	Well-balanced orientations	34
3.1.2	Orientations for local arc-connectivity	43
3.1.3	Orientations for arc-connectivity and parity	46
3.1.4	Orientations of graphs of odd edge-connectivity	57
3.2	Orientations for vertex-connectivity	73
3.2.1	Finding one highly vertex-connected orientation	73
3.2.2	Graphs all of whose Eulerian orientations are highly vertex-connected	75
4	Arborescence packing	91
4.1	Introduction to arborescence packing	91
4.1.1	Packing of spanning dyperarborescences	93
4.1.2	Packing of mixed arborescences	93
4.1.3	Matroid-based packings	94
4.1.4	Arborescence packings and matroid intersection	94
4.1.5	Algorithmic aspects	95
4.2	Reachability in arborescence packings	96
4.2.1	Basic setting	97
4.2.2	Matroid-reachability-based arborescence packings	100
4.2.3	Reachability in packings of mixed hyperarborescences	103
4.2.4	Algorithmic aspects	108

4.3	Packings of mixed hyperarborescences with flexible roots	110
4.3.1	Previous work	110
4.3.2	Matroid intersection	111
4.3.3	Proof of Theorem 4.3.3	114
4.3.4	Algorithmic aspects	116
4.4	Packing k -safe arborescences and related objects	117
4.4.1	Preliminaries	120
4.4.2	An FPT algorithm for packing k -safe spanning arborescences	122
4.4.3	An FPT algorithm for packing spanning (r, k) -flow branchings	125
4.4.4	An FPT algorithm for packing (r, k) -safe spanning trees	127
4.4.5	A hardness result for packing (r, k) -safe spanning trees	130
4.4.6	Conclusion	132
5	Connectivity augmentation	133
5.1	Edge-, arc- and vertex-connectivity augmentation	133
5.2	Mixed connectivity augmentation	135
5.2.1	Basic definitions and previous results	136
5.2.2	Minimal even extensions for $(2, k)$ -connectivity	140
5.2.3	A fast splitting off algorithm	142
6	Conclusion	148

Acknowledgements

I first wish to thank the two reviewers of my thesis, Tibor Jordán and Matthias Kriesell. They both invested a great amount of time in reading my thesis which led to a significant list of deep insights which allowed me to improve the quality of this thesis. I also wish to thank Matthias Kriesell for being ready to spend some more time with me and giving me the possibility to continue my research with him in my beautiful motherland Germany.

I also wish to thank the other members of the jury, Nadia Brauner, Frédéric Havet and Stéphan Thomassé, who made all the way to Grenoble for my defence. Their questions and ideas have been a great source of inspiration of me.

I now wish to acknowledge the person which has by far had the largest influence on this work, my supervisor Zoltán Szigeti. It is hard to describe in a few words how huge his support was during all this time. While other PhD students would meet their supervisor once a week, I never had to schedule a meeting and would just knock on his door to discuss, normally several times per day. I remember many weeks in which I spent more time in his office than in my own. Not even the Covid crisis was able to interrupt this close collaboration as we sent messages almost every day and Zoltán always found time when I wanted to talk to him.

Due to the immense depth of his knowledge on graph theory, I have been able to learn many things from him. I am still impressed by the sharpness of his knowledge and the enormous rigour he applied when dealing with my suggestions and proofreading the drafts I gave him.

Further, he took a lot of time for teaching me how to pronounce hungarian names, in particular his own. Finally, he constantly supplied me with sweets, most often desserts from the Indian restaurant, an effort that clearly goes beyond the usual duties of a supervisor!

Pour tout cela, merci maître!

I further wish to thank Stéphane Bessy. First, he volunteered as an expert for my CSI and gave valuable hints for my thesis. After, he invited me to Montpellier. Even though we could not meet in person due to the Covid crisis, his invitation led to fruitful collaboration, in particular with him and Ignasi Sau whom I also wish to thank.

A PhD can hardly ever go well without a helpful environment. This environment was provided by the GSCOP lab which I really enjoyed being part of during these three years. Whenever there were any kind of problems, scientific, technical or personal, I always knew I could knock on any door and the persons inside would do whatever they could to help me.

Plus particulièrement, j'aimerais remercier Nadia et Moritz chaleureusement pour m'avoir donné la possibilité d'enseigner avec eux. J'ai beaucoup appris de leur manière d'enseigner et j'espère un jour pouvoir enseigner aux étudiants comme eux. L'administration a toujours posé un défi pour moi pendant ces trois ans. Heureusement, j'étais avec Cléopée, une excellente cobureau et sœur de thèse qui s'est occupée d'un nombre énorme

de documents. Dans les rares moments quand elle ne savait pas quoi faire, j'ai toujours trouvé de l'aide chez Fadila ou Marie-Jo, la maman des doctorants. Merci! Je veux aussi remercier Kevin et Olivier qui ont fait des efforts significatifs pour construire un pont entre la technologie moderne et moi. En plus, je veux passer un grand merci à Margaux qui a amélioré mes connaissances minables de Latex et à Louis qui m'a introduit à IPE. Finalement, je veux remercier tout le monde pour avoir passé du temps à relire mes textes en français, y compris ces remerciements. Grace à vous, j'ai pu atteindre une maîtrise modeste de cette belle langue!

Ma thèse n'aurait pas été pareille si je n'avais pas pu partager mon temps libre (et le temps pendant lequel j'étais censé travailler mais ne l'ai pas fait) avec des gens qui sont devenus de bons amis durant ces trois ans. Je n'oublierai jamais ces moments magnifiques qu'on a passés au cours des sorties à la montagne, des sorties piscines, des barbecs ou juste en prenant un café ou une petite bière. Je veux remercier Cléopée pour m'avoir invité à sa merveilleuse maison au Lauzet, Maud pour les énormes soirées à Brignoud, Astor pour m'avoir appris à jouer au Go, Margaux et Pascal pour m'avoir appris à jouer à la coinche et Valentin, mon frère de thèse, pour nos parties de billard. I wish to thank Akash and Marco for the great hikes in the beautiful nature around Grenoble we went for together and I wish to acknowledge Bilge and Asiye for preparing delicious turkish food at many occasions.

L'espace ici n'est pas suffisant pour remercier tous les gens que j'ai rencontré à Grenoble de la manière qu'ils auraient méritée, je désire donc juste dire en plus un grand merci aux gens suivants pour m'avoir permis de passer du temps avec eux: Victor, Cléa, les Valentins, Moritz, Alex, Tamara, Tatiana, Florence, Camille, Florian, Nico, Luc, Thomas, Léa, Gricha, Lucie, Hugo, Ugo, Lucas, Ernest, Zakaria, Maria, ...

Je veux aussi remercier Nico du Sun Valley qui nous a toujours accueilli pour une petite IPA et le club de ping-pong de St. Egrève qui à coté du sport était un club énormément convivial.

Ich möchte diesen letzten Abschnitt meiner Familie widmen. Sie haben mich während der langen Zeit immer unterstützt. Ich möchte meinen Eltern dafür danken, dass sie mich in der langen Zeit in Grenoble mehrfach besucht haben, um mich zum Essen einzuladen und mein Fahrrad technisch in Schuss zu halten. Als ich während der Coronakrise einige Zeit in Deutschland verbrachte, wurde ich bei ihnen wieder aufgenommen und konnte mich sofort wieder zuhause fühlen. Dafür möchte ich meinen Eltern, aber auch meinen Schwestern, Großeltern und Freunden in Deutschland herzlich danken.

Chapter 1

Introduction

Networks are a structure humanity had to deal with during many stages of its development. Historically, the task was to represent the parental relationships between the greek gods, to decide about the distribution of consuls to provinces in ancient Rome or to plan the tours of ships during early colonialism. Nowadays, we wish to understand the structure of a social network, to model the transportation tasks of a big logistics company or to send information as fast as possible through a modern computer network. Despite their diversity, all of these problems have an important feature in common: Their underlying structure can be modeled by a set of points some of which are joined by a link. Such a structure is called a *network*. One of the most important notions in the analysis of networks is connectivity. The first important question in the context of connectivity is the following one: Travelling along the links, can we reach every point from every other point? Another more general question is whether we can still do so if our network is damaged meaning that some points or links are eliminated from it.

In modern mathematics and computer science, there are two major ways of modeling networks. The first one is called *undirected graphs*. In undirected graphs the points of the network are represented by a ground set called the set of vertices and the links are represented by a set of so-called edges. Every edge connects two vertices and it is symmetric in the sense that it can be used to travel in either of the two directions. For example, a road network not containing any one-way street can be modeled using undirected graphs. The second model is called *directed graphs*. Again, the points of the network are modeled by a set of vertices. The difference now is that instead of edges we have so-called *arcs* where an arc goes from one vertex to another vertex and can only be travelled in that one direction. For example, directed graphs can be used to model a network of inland waterway transportation where the flow direction of rivers plays a significant role.

While we postpone the formal definitions of graph connectivity to Chapter 2, we wish to mention that there are two major notions for connectivity in undirected graphs. The notion of *edge-connectivity* describes how stable the graph is with respect to the elimination of some edges and the notion of *vertex-connectivity* describes how stable the graph is with respect to the elimination of some vertices. For directed graphs, we similarly distinguish between *arc-connectivity* and *vertex-connectivity*. Despite the historical applications described earlier, it was only in the earlier 20th century that graph connectivity was studied on high scientific level. In 1927, Menger [93] provided a collection of results for all kinds of connectivity which, until today, can be considered the most fundamental and influential results there are in graph connectivity.

Since then, connectivity has grown to be one of the most important notions of graph theory in general. While connectivity serves as an important tool in a wide range of both

structural and algorithmic fields of graph theory, there are also a lot of problem settings which are more purely related to connectivity. While also this field is too immense to be surveyed in a single work, the book of Frank [41] gives an excellent overview of some of the most important problems in this field. Many of the problems considered in this thesis have been brought to our attention through this book. The goal of this thesis is to enrich the theory of graph connectivity by some new results. In the main part of this thesis, more specifically Chapters 3, 4 and 5, we deal with three more specific classes of problems related to connectivity. In Chapter 2, we give some more formal definitions and some preliminary results. We now give a more close description of the problems we deal with in the 3 main chapters.

In Chapter 3, we deal with a class of problems which is located at the intersection of undirected and directed connectivity. An *orientation* of an undirected graph is a directed graph which is obtained by replacing every edge by an arc between the same two vertices. We wish to examine whether certain connectivity properties of the original undirected graph can be maintained through that orientation operation. The first important result in this field is due to Robbins [100] and dates back to 1939.

In the light of the result of Robbins and a result of Nash-Williams from 1960 [96], the possibilities of connectivity maintenance are relatively well understood for the case when some edge-connectivity property in the undirected graph is supposed to yield an arc-connectivity property in its orientation. Nevertheless, several refinements of these results can be considered some of which we deal with in Section 3.1. In particular, there is a much stronger version of the theorem of Nash-Williams. We deal with some results related to this stronger theorem. In particular, we give a negative answer to a question of Frank [41]. Further, we deal with some problems aiming to achieve some slightly more sophisticated connectivity notions in the orientation. The first such connectivity notion is defined by some local connectivity condition. Again, we give a negative result. Both these negative results can also be found in [59]. We next consider a further orientation problem that establishes a link between the connectivity of the orientation and a parity consideration. Finally, we deal with a setting where we wish to achieve a connectivity property in the orientation that is in some way in the middle between two well-known arc-connectivity properties. This leads to the definition a new graph invariant. We provide several bounds on the possible values of this invariant. This is based on joint work with Szigeti [60] in response to a suggestion of Frank [45].

Much less is known for the case where a vertex-connectivity property in the undirected graph is supposed to yield a vertex-connectivity property in its orientation. While a theorem of Thomassen [105] implies a good characterization in a basic case, a result of Durand de Gevigney [24] makes a similar result seem to be out of reach in the general case. Nevertheless, many questions in this field remain wide open. In Section 3.2, we give an overview of these results. Next, we deal with a slightly different problem suggested by Cheriyan [19]. We show that for some restricted classes of graphs, all orientations with a certain extra property are highly vertex-connected. This is based on joint work with Szigeti [61].

In Chapter 4, we deal with a set of problems in directed graphs. More precisely, we here wish to decompose a directed graph into a maximum number of subgraphs all of which have a certain connectivity property. Graphs that minimally have this property are called *spanning arborescences* and we can restate this problem as the one of finding a maximum number of arc-disjoint spanning arborescences. In the most basic setting, this problem is solved thanks to a fundamental theorem of Edmonds [28]. We deal with 3 generalizations of this theorem. In the first two generalizations we somewhat relax the connectivity conditions imposed on the subgraphs in certain ways. Firstly, we show that results for a certain relaxation of this condition can be derived from the results in the

original setting avoiding earlier, more involved proofs. This is based on joint work with Szigeti [63]. Secondly, we extend a result of Gao and Yang [55] on packing spanning mixed arborescences whose roots are not fixed to a more general setting. Again, this is joint work with Szigeti [62]. In order to obtain this result, we rely on another important object in combinatorial optimization: matroids. Finally, we deal with another generalization of the theorem of Edmonds in which some additional conditions are imposed on the spanning arborescences. Our approach works for spanning arborescences and two similar objects. These results have been obtained in joint work with Bessy, Maia, Rautenbach and Sau [16] and give and yield an answer to a question of Bang-Jensen, Havet and Yeo [6].

In Chapter 5, we deal with connectivity augmentation problems meaning we want to add a minimum number of edges to an undirected graph or arcs to a directed graph in order to obtain a certain connectivity property. We give an overview of the results for this problem which is well understood for all basic kinds of connectivity. We add a new result for a more advanced notion of connectivity. This is joint work with Szigeti [64].

Chapter 2

Preliminaries

In this section, we give a collection of definitions and basic results we require in this thesis. In Section 2.1, we formally introduce the objects that play the most important role in this thesis: graphs and similar objects like directed graphs and some more general objects. We give some of the basic results in graph theory. In Section 2.2, we deal with two slightly more sophisticated tools in graph theory, namely splitting off and submodularity. In Section 2.3, we consider relevant algorithmic notions. In particular, we deal with several complexity classes. Section 2.4 deals with a different object that also plays a significant role in this thesis: matroids. We introduce some basics of matroids and describe a few particular matroids.

2.1 Graphs

In this section, we formally introduce graphs and related objects. We first give some basic notions. After, we deal with some basic operations that can be applied to graphs. Finally, we introduce some particular graphs that play an important role in this thesis.

2.1.1 Basic notions

We first go through the basic notions for sets and bisets. After, we deal with undirected and directed graphs and mixed hypergraphs.

2.1.1.1 Sets and bisets

Most sets considered in this thesis are finite. We consider that a *set* contains no identical elements, otherwise it is called a *multiset*. We use \emptyset to denote the set not containing any element, also called the *empty* set. Any set that contains an element is called *nonempty*. If a set contains a single element x , we often use \mathbf{x} instead of $\{x\}$. Given two sets S_1 and S_2 , we say that S_1 is a *subset* of S_2 and write $\mathbf{S}_1 \subseteq \mathbf{S}_2$ if every element of S_1 is also in S_2 . If additionally $S_1 \neq S_2$, we say that S_1 is a *proper* subset of S_2 and write $S_1 \subsetneq S_2$. We define the *intersection* of S_1 and S_2 by $\mathbf{S}_1 \cap \mathbf{S}_2 = \{x : x \in S_1 \text{ and } x \in S_2\}$ and the *union* of S_1 and S_2 by $\mathbf{S}_1 \cup \mathbf{S}_2 = \{x : x \in S_1 \text{ or } x \in S_2\}$. We say that S_1 and S_2 are *disjoint* if $S_1 \cap S_2 = \emptyset$. We further define $\mathbf{S}_1 - \mathbf{S}_2 = \{x : x \in S_1 \text{ and } x \notin S_2\}$. We say that S_1 and S_2 are *properly intersecting* if none of $S_1 - S_2$, $S_2 - S_1$ and $S_1 \cap S_2$ is empty. We say that S_1 *separates* S_2 if both $S_1 \cap S_2$ and $S_2 - S_1$ are nonempty. Next, we denote by $\mathbf{S}_1 \times \mathbf{S}_2$ the set $\{(s_1, s_2) : s_1 \in S_1, s_2 \in S_2\}$. We abbreviate $S_1 \times S_1$ to S_1^2 . Given a set S , we call a collection of disjoint subsets of S a *subpartition* of S . If in addition, the union of these subsets is S , we speak of a *partition* of S . We use $\mathbf{2}^S$ for the power set of S , i.e. the set

that contains all subsets of S . For a function $f : S_1 \rightarrow S_2$ and $X \subseteq S_1$, we use $\mathbf{f}[\mathbf{X}]$ for the restriction of f to X . A set that contains at least two elements is called *nonsingular*.

For some $x \in S_1$ and $y \notin S_1$, we say that S_1 is an $x\bar{y}$ -set.

We use \mathbb{R} for the set of real numbers, \mathbb{Z} for the set of integers and $\mathbb{Z}_{\geq 0}$ for the set of nonnegative integers. For some $\alpha \in \mathbb{R}$, we use $\lfloor \alpha \rfloor$ to denote the biggest integer that is not bigger than α and $\lceil \alpha \rceil$ to denote the smallest integer that is not smaller than α .

We now get to a somewhat less common notion that was introduced by Frank and Jordán in [47], namely bisets. This notion will prove useful in several parts of this thesis. Given a ground set Ω , a *biset* \mathbf{X} on Ω is a tuple (X_I, X_O) of subsets of Ω , called the *inner set* and the *outer set* of Ω such that $X_I \subseteq X_O \subseteq \Omega$. A biset is called *trivial* if $X_I = \emptyset$ or $X_O = \Omega$, *nontrivial* otherwise. More generally, given some $\Omega' \subseteq \Omega$, we say that \mathbf{X} is *nontrivial with respect to Ω'* if $X_O \subsetneq \Omega'$ and $X_I \neq \emptyset$.

We define the complement $\bar{\mathbf{X}}$ of \mathbf{X} by $\bar{X}_I = \Omega - X_O$ and $\bar{X}_O = \Omega - X_I$. We denote $X_O - X_I$ by $\mathbf{w}(\mathbf{X})$, the *wall* of \mathbf{X} . Given two bisets \mathbf{X} and \mathbf{Y} , we define the *union* $\mathbf{X} \cup \mathbf{Y}$ by $(\mathbf{X} \cup \mathbf{Y})_I = X_I \cup Y_I$ and $(\mathbf{X} \cup \mathbf{Y})_O = X_O \cup Y_O$. Further, we define the *intersection* $\mathbf{X} \cap \mathbf{Y}$ by $(\mathbf{X} \cap \mathbf{Y})_I = X_I \cap Y_I$ and $(\mathbf{X} \cap \mathbf{Y})_O = X_O \cap Y_O$. Further, \mathbf{X} and \mathbf{Y} are called *innerly disjoint* if $X_I \cap Y_I = \emptyset$.

A schematic drawing can be found in Figure 2.1.1

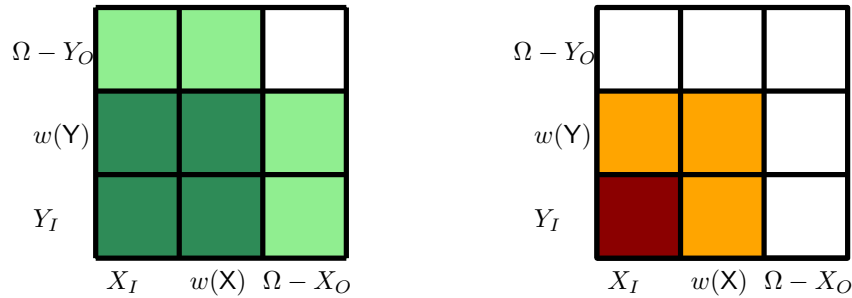


Figure 2.1.1: A schematic drawing of the union and the intersection of bisets. On the left side, $(\mathbf{X} \cup \mathbf{Y})_I$ is depicted in dark green while $w(\mathbf{X} \cup \mathbf{Y})$ is depicted in light green. On the right side, $(\mathbf{X} \cap \mathbf{Y})_I$ is depicted in red while $w(\mathbf{X} \cap \mathbf{Y})$ is depicted in orange.

For a set $S \subseteq \Omega$, a collection of bisets $\{\mathcal{X}^i\}_1^\ell$ is called a *biset subpartition* of S if $\{X_I^i\}_1^\ell$ is a subpartition of S and $w(\mathcal{X}^i) \subseteq \Omega - S$ for $i = 1, \dots, \ell$.

2.1.1.2 Undirected graphs

An *undirected graph* is a tuple $G = (V, E)$ where V is a set called *vertex set*. Further, E is a multiset such that every $e \in E$ is a multisubset of V of size exactly 2. While infinite graphs are an active field of research (see e.g. [82]), all graphs considered in this thesis will be finite. Often, we abbreviate an undirected graph to a *graph*. We abbreviate an edge $e = \{u, v\}$ to uv . We say that u and v are the *ends* or *endvertices* of e and that e is *between* u and v or *connects* u and v . Further, we say that u and v are *adjacent* to each other and *incident* to e . We use \mathbf{n} and \mathbf{m} for the number of vertices and edges of a given graph, respectively. Also, usually, a graph is illustrated by drawing a point for every vertex and a line between the two points that correspond to its endvertices for every edge. An example can be found in Figure 2.1.2.

For some $X \subseteq V$, we denote $\mathbf{E}(\mathbf{X}) = \{e \in E : e \subseteq X\}$. A *subgraph* of G is a graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. For some $v \in V$, we use $\mathbf{N}_G(\mathbf{v}) = \{u \in V - v : uv \in E\}$ and we call this set the set of *neighbors* of v . For some $X \subseteq V$, we denote by $\mathbf{G}[\mathbf{X}]$ the subgraph of G that is *induced* on X , i.e. the graph whose vertex set is X and whose edge set contains all the edges of E both of whose endvertices are

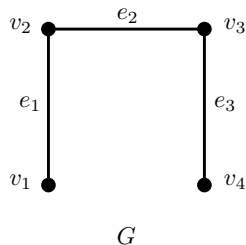


Figure 2.1.2: An example for a drawing of a graph $G = (V, E)$ where $V = \{v_1, v_2, v_3, v_4\}$ and $E = \{e_1 = v_1v_2, e_2 = v_2v_3, e_3 = v_3v_4\}$

in X . We abbreviate $G[V - X]$ to $\mathbf{G} - \mathbf{X}$. We use $\mathbf{i}_G(\mathbf{X})$ for $|E(G[X])|$. Further, for an edge $e \in E$, we use $\mathbf{G} - e$ for $(V, E - e)$. For a new edge $e \notin E$, we use $\mathbf{G} \cup e$ for $(V, E \cup e)$. An edge $e \in E$ which contains twice the same element is called a *loop*. Two edges with the same endvertices are called *parallel*. A graph that does not contain loops or parallel edges is called *simple*. Two graphs are called *edge-disjoint* if their edge sets are disjoint. For two graphs $G_1 = (V, E_1), G_2 = (V, E_2)$ on the same vertex set V , we use $\mathbf{G}_1 \cup \mathbf{G}_2$ for $(V, E_1 \cup E_2)$. A *singularly rooted graph* is a graph $G = (V \cup r, E)$ with a specified vertex r called the *root*.

Given some $X \subseteq V$, we denote by $\delta_G(\mathbf{X})$ the set of edges that have exactly one end in X and we use $\mathbf{d}_G(\mathbf{X})$ for $|\delta_G(\mathbf{X})|$. For a single vertex $v \in V$, we call $\mathbf{d}_G(v)$ the *degree* of v . For $X, Y \subseteq V$, we use $\delta_G(\mathbf{X}, \mathbf{Y})$ for the set of edges that have one end in $X - Y$ and one end in $Y - X$. Further, we denote the number of edges that have one end in $X \cap Y$ and one end in $V - (X \cup Y)$ by $\bar{\delta}_G(\mathbf{X}, \mathbf{Y})$. We use $\mathbf{d}_G(\mathbf{X}, \mathbf{Y})$ and $\bar{\mathbf{d}}_G(\mathbf{X}, \mathbf{Y})$ for $|\delta_G(\mathbf{X}, \mathbf{Y})|$ and $|\bar{\delta}_G(\mathbf{X}, \mathbf{Y})|$, respectively. A graph all of whose vertices are of degree k for some $k \in \mathbb{Z}_{\geq 0}$ is called *k-regular*. We abbreviate 3-regular to *cubic*. Given some $k \in \mathbb{Z}_{\geq 0}$, we say that G is *k-edge-connected* if $\mathbf{d}_G(X) \geq k$ for all nonempty, proper subsets X of V . We abbreviate 1-edge-connected to *connected*. For $x, y \in V$, we use $\lambda_G(x, y)$ for $\min\{\mathbf{d}_G(X) : X \text{ is an } x\bar{y}\text{-set}\}$. We next give a less common notion which will be used in Section 3.1.4. We say that G is *essentially k-edge-connected* if G is $(k - 1)$ -edge-connected and $\mathbf{d}_G(X) \geq k$ for every $X \subseteq V$ with $|X|, |V - X| \geq 2$. We further say that G is *k-vertex-connected* if $|V| \geq k + 1$ and $G - X$ is connected for every $X \subseteq V$ with $|X| \leq k - 1$. If G is connected, we say that some $v \in V$ is a *cutvertex* if $G - v$ is not connected. We further give a mixed notion of edge-connectivity and vertex-connectivity that was introduced by Kaneko and Ota in [72]. Given integers $\ell, k \in \mathbb{Z}_{\geq 0}$, we say that G is (ℓ, k) -connected if $|V| \geq \ell + 1$ and $G - X$ is $k(\ell - |X|)$ -edge-connected for every $X \subseteq V$ with $|X| \leq \ell - 1$. For some $k \in \mathbb{Z}_{\geq 0}$, a *coloring* of V is a function $\phi : V \rightarrow \{1, \dots, k\}$. A coloring is called *proper* if $\phi(u) \neq \phi(v)$ for all u, v with $uv \in E$. We call k the *number of colors*. A graph that has a proper coloring with two colors is called *bipartite*. The *line graph* of G is the graph whose vertex set is E and that contains an edge between two vertices $e, f \in E$ if $e \cap f \neq \emptyset$.

A *capacitated graph* is a graph $G = (V, E)$ where E does not contain parallel elements together with a capacity function $c : E \rightarrow \mathbb{Z}_{\geq 0}$. Clearly, a capacitated graph can be transformed into a graph by replacing every edge of a certain capacity by the corresponding number of parallel edges and vice-versa. However, when dealing with the running times of algorithms, the distinction is crucial.

2.1.1.3 Directed graphs

A *directed graph* is a tuple $D = (V, A)$ where V is a set called *vertex set*. Further, A is a multiset called the set of *arcs* of D where every $a \in A$ is an element of V^2 . The first element of a is called the *tail* and the second element is called the *head* of a . Often, we

abbreviate a directed graph to a *digraph*. We abbreviate an arc $a = (u, v)$ to uv with $\mathbf{tail}(a) = u$ and $\mathbf{head}(a) = v$. We also say that a goes from u to v . We use n and m for the number of vertices and arcs of a given digraph, respectively. Usually, a digraph is illustrated by drawing a point for every vertex and an arrow from the point corresponding to its tail to the point corresponding to its head for every arc. An example can be found in Figure 2.1.3.

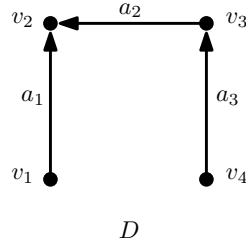


Figure 2.1.3: An example for a drawing of a digraph $D = (V, A)$ where $V = \{v_1, v_2, v_3, v_4\}$ and $A = \{a_1 = v_1v_2, a_2 = v_3v_2, a_3 = v_4v_3\}$.

For some $X \subseteq V$, we denote $\mathbf{A}(X) = \{a \in A : \mathbf{tail}(a), \mathbf{head}(a) \in X\}$. A *subgraph* of D is a digraph $D' = (V', A')$ such that $V' \subseteq V$ and $A' \subseteq A$. For some $X \subseteq V$, we denote by $\mathbf{D}[X]$ the subgraph of D that is *induced* on X , i.e. the graph whose vertex set is X and whose arc set contains all the arcs of A whose head and whose tail is in X . We abbreviate $\mathbf{D}[V - X]$ to $\mathbf{D} - X$. An arc $a \in A$ with $\mathbf{head}(a) = \mathbf{tail}(a)$ is called a *loop*. Two identical arcs are called *parallel*. A digraph without loops or parallel arcs is called *simple*. Two digraphs are *vertex-disjoint* (*arc-disjoint*) if their vertex sets (arc sets) are disjoint. For some $v \in V$, we use $\mathbf{N}_D^-(v) = \{u \in V : uv \in A\}$ and $\mathbf{N}_D^+(v) = \{u \in V : vu \in A\}$ and we call these sets the set of *inneighbors* and *outneighbors* of v , respectively. For two digraphs $D_1 = (V, A_1), D_2 = (V, A_2)$ on the same vertex set V , we use $\mathbf{D}_1 \cup \mathbf{D}_2$ for $(V, A_1 \cup A_2)$.

Given some $X \subseteq V$, we denote by $\delta_D^-(X)$ the set of arcs whose tail is in $V - X$ and whose head is in X . We use $d_D^-(X)$ for $|\delta_D^-(X)|$, $\delta_D^+(X)$ for $\delta_D^+(V - X)$ and $d_D^+(X)$ for $|\delta_D^+(X)|$. For a single vertex $v \in V$, we call $d_D^-(v)$ and $d_D^+(v)$ the *in-degree* and *out-degree* of v , respectively. Given some $k \in \mathbb{Z}_{\geq 0}$, we say that D is *k-arc-connected* if $d_D^-(X) \geq k$ for all nonempty, proper subsets X of V . We use *strongly connected* for 1-arc-connected. For $x, y \in V$, we use $\lambda_D(x, y)$ for $\min\{d_D^+(X) : X \text{ is an } x\bar{y}\text{-set}\}$. We further say that D is *k-vertex-connected* for some $k \in \mathbb{Z}_{\geq 0}$ if $|V| \geq k + 1$ and $\mathbf{D} - X$ is strongly connected for every $X \subseteq V$ with $|X| \leq k - 1$. A vertex-maximal strongly connected induced subgraph of D is called a *strongly connected component* of D . The following simple result is well-known.

Proposition 2.1.1. *Let $D = (V, A)$ be a digraph. There is some $X \subseteq V$ such that $\mathbf{D}[X]$ is a strongly connected component of D and $d_D^+(X) = 0$.*

The *underlying graph* of a digraph is obtained by replacing every arc by an edge connecting its head and its tail. We say that a digraph is *weakly connected* if its underlying graph is connected.

A vertex $v \in V$ is called a *sink* if $d_D^+(v) = 0$. Next, a vertex $r \in V$ is called a *root* if $d_D^-(r) = 0$. Further, r is called a *simple root* if $d_D^+(r) \leq 1$. A (*simply*) *rooted digraph* is a digraph $D = (V \cup R, A)$ where R is a set of (simple) roots. If R contains a single vertex, we say that D is *singularly rooted*. A singularly rooted digraph $D = (V \cup r, A)$ such that $d_D^-(X) \geq k$ for all nonempty $X \subseteq V$ for some $k \in \mathbb{Z}_{\geq 0}$ is called *k-root-connected*. We abbreviate 1-root-connected to *root-connected*.

2.1.1.4 Mixed hypergraphs

In this section we introduce a much more general object which is called mixed hypergraphs and generalizes both graphs and digraphs.

A *mixed hypergraph* is a tuple $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ where V is a set of vertices. Next, \mathcal{A} is a set of directed hyperedges (dyperedges), i.e. every $a \in \mathcal{A}$ is a tuple $(tail(a), head(a))$ where $head(a)$ is a vertex in V and $tail(a)$ is a nonempty subset of $V - head(a)$. Further, \mathcal{E} is a set of hyperedges, i.e. every $e \in \mathcal{E}$ is a subset of V of size at least 2. A mixed hypergraph without hyperedges is called a *directed hypergraph (dypergraph)* while a mixed hypergraph without dyperedges is called a *hypergraph*. We say that \mathcal{F} is a mixed graph if each dyperedge has a tail of size exactly one and each hyperedge contains exactly two vertices. For some $a \in \mathcal{A}$, its *underlying hyperedge* is $head(a) \cup tail(a)$. The *underlying hypergraph* of \mathcal{F} is obtained by replacing every $a \in \mathcal{A}$ by its underlying hyperedge.

Let $X \subseteq V$. We say that dyperedge $a \in \mathcal{A}$ *enters* X if $head(a) \in X$ and $tail(a) - X \neq \emptyset$ and a *leaves* X if a enters $V - X$. We denote by $\delta_{\mathcal{A}}^-(X)$ the set of dyperedges entering X and by $\delta_{\mathcal{A}}^+(X)$ the set of dyperedges leaving X . We use $d_{\mathcal{A}}^-(X)$ for $|\delta_{\mathcal{A}}^-(X)|$ and $d_{\mathcal{A}}^+(X)$ for $|\delta_{\mathcal{A}}^+(X)|$. We say that a hyperedge e *enters* or *leaves* X if e intersects both X and $V - X$ and denote by $d_{\mathcal{E}}(X)$ the number of hyperedges entering X . We call a vertex r a *root* in \mathcal{F} if $d_{\mathcal{A}}^-(r) = d_{\mathcal{E}}(r) = 0$ and $tail(a) = \{r\}$ for all $a \in \delta_{\mathcal{A}}^+(r)$ and a *simple* root if additionally $d_{\mathcal{A}}^+(r) \leq 1$. Given a subpartition $\{V_i\}_1^\ell$ of V and some $\mathcal{A}' \cup \mathcal{E}' \subseteq \mathcal{A} \cup \mathcal{E}$, we denote by $e_{\mathcal{A}' \cup \mathcal{E}'}(\{V_i\}_1^\ell)$ the number of dyperedges and hyperedges in $\mathcal{A}' \cup \mathcal{E}'$ entering some V_i ($i \in \{1, \dots, \ell\}$). A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is called *partition-connected* if $e_{\mathcal{E}}(\mathcal{P}) \geq |\mathcal{P}| - 1$ for every partition \mathcal{P} of V . A dypergraph $\mathcal{D} = (V, \mathcal{A})$ is called *strongly connected* if $d_{\mathcal{A}}^-(X) \geq 1$ for every nonempty $X \subsetneq V$. Further, we call \mathcal{D} *k-dyperedge-connected* if the deletion of any $k - 1$ dyperedges from \mathcal{D} leaves a strongly connected dypergraph. For some $\mathcal{A}' \cup \mathcal{E}' \subseteq \mathcal{A} \cup \mathcal{E}$, we use $V(\mathcal{A}' \cup \mathcal{E}')$ for $\bigcup_{a \in \mathcal{A}'} tail(a) \cup head(a) \bigcup_{e \in \mathcal{E}'} e$.

Trimming a dyperedge a means that a is replaced by an arc uv with $v = head(a)$ and $u \in tail(a)$. *Trimming* a hyperedge e means that e is replaced by an arc uv for some $u \neq v \in e$.

We define a (*simply*) *rooted mixed hypergraph* as a mixed hypergraph $\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E})$ with R being a set of (simple) roots.

A mixed hypergraph \mathcal{F} is called *grounded* if $\mathcal{A} = \delta_{\mathcal{F}}^+(R)$. The *underlying grounded mixed hypergraph* of a mixed hypergraph \mathcal{F} is obtained by replacing all dyperedges in $\mathcal{A} - \delta_{\mathcal{F}}^+(R)$ by the corresponding hyperedges. For a hyperedge $e \in \mathcal{E}$, its corresponding *bundle* \mathcal{A}_e is the set of all possible orientations of e , i.e. $\mathcal{A}_e = \{(e - v, v) : v \in e\}$. The *directed extension* $\mathcal{D}_{\mathcal{F}} = (V \cup R, \mathcal{A} \cup \mathcal{A}_{\mathcal{E}})$ of \mathcal{F} is obtained by replacing every $e \in \mathcal{E}$ by its corresponding bundle.

2.1.2 Operations on graphs and digraphs

In this section, we discuss two operations that can be applied to graphs, namely contractions and orientations.

2.1.2.1 Contraction

First let $G = (V, E)$ be an undirected graph and $X \subseteq V$. We now define a graph $G' = (V', E')$ where $V' = V - X \cup x$ for a new vertex x . Next, E' contains every edge $f = yz \in E$ with $\{y, z\} \cap X = \emptyset$. Further, for every edge yz with $y \in V - X$ and $z \in X$, E' contains the edge yx . Observe that all edges having both ends in X do not appear in G' . We say that G' is obtained from G by *contracting* X and write $G' = \mathbf{G}/\mathbf{X}$. For some subgraph H of G , we abbreviate $G/(V(H))$ to \mathbf{G}/\mathbf{H} . For a collection $\mathcal{H} = \{H_1, \dots, H_t\}$ of vertex-disjoint subgraphs of G , we use \mathbf{G}/\mathcal{H} for $((G/H_1)/\dots)/H_t$. An example can be found in Figure 2.1.4.

For a vertex $v \in V$ of degree 2, observe that when the set containing v and one of its two neighbors is contracted, twice the same graph arises. We call this the operation the *suppression* of v .

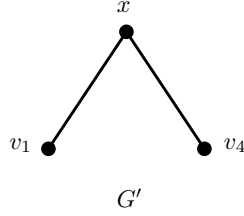


Figure 2.1.4: The graph G' is obtained from G in Figure 2.1.2 by contracting $\{v_2, v_3\}$ into x .

Next, let $D = (V, A)$ be a directed graph and $X \subseteq V$. We now define a digraph $D' = (V', A')$ where $V' = V - X \cup x$ for a new vertex x . First, A' contains every arc $f = yz \in A$ with $\{y, z\} \cap X = \emptyset$. Further, for every arc $yz \in A$ with $y \in V - X$ and $z \in X$, A' contains the arc yx and for every arc $yz \in A$ with $y \in X$ and $z \in V - X$, A' contains the arc xz . Observe that all arcs with head and tail in X do not appear in D' . We say that D' is obtained from *contracting* X and write $D' = \mathbf{D}/\mathbf{X}$. For some subgraph Q of D , we abbreviate $D/(V(Q))$ to \mathbf{D}/\mathbf{Q} .

We now give a collection of easy results that show that certain connectivity properties are maintained through contractions.

Proposition 2.1.2. *Let $G = (V, E)$ be a graph, $X \subseteq V$ and $k \in \mathbb{Z}_{\geq 0}$.*

- (a) *If G is k -edge-connected, then so is G/X .*
- (b) *If G is essentially k -edge-connected, then so is G/X .*

Proof. (a): Let \mathbf{x} be the vertex X is contracted to and consider some $\mathbf{S} \subseteq V - X \cup x$. By symmetry, we may suppose that $x \in (V - X \cup x) - \mathbf{S}$. We then obtain $d_{G/X}(\mathbf{S}) = d_G(\mathbf{S}) \geq k$, so G/X is k -edge-connected.

(b): Again, let \mathbf{x} be the vertex X is contracted to. By (a) and as G is $(k - 1)$ -edge-connected, we obtain that G/X is $(k - 1)$ -edge-connected. Now consider some $\mathbf{S} \subseteq V - X \cup x$ with $|\mathbf{S}|, |(V - X \cup x) - \mathbf{S}| \geq 2$. By symmetry, we may suppose that $x \in (V - X \cup x) - \mathbf{S}$. As G is essentially k -edge-connected and $|V - \mathbf{S}| \geq |(V - X \cup x) - \mathbf{S}| \geq 2$, we obtain $d_{G/X}(\mathbf{S}) \geq d_G(\mathbf{S}) \geq k$. It follows that G/X is essentially k -edge-connected. \square

Proposition 2.1.3. *For a subgraph Q of a directed graph $D = (V, A)$, we have*

- (a) *if D is strongly connected, then so is D/Q ,*
- (b) *if D/Q and Q are strongly connected, then so is D .*

Proof. The proof of (a) is similar to the one of Proposition 2.1.2 (a). For (b), let $X \subseteq V$. If $X \cap V(Q)$ and $X \cap (V - V(Q))$ are nonempty, as Q is strongly connected, we obtain $d_D^-(X) \geq d_Q^-(X \cap V(Q)) \geq 1$. We may hence suppose by symmetry that $X \cap V(Q) = \emptyset$. In that case, as D/Q is strongly connected, we obtain $d_D^-(X) = d_{D/Q}^-(X) \geq 1$. In any case, we have $d_D^-(X) \geq 1$, so D is strongly connected. \square

2.1.2.2 Orientation

We now give the definition of an operation which will play a major role in this thesis. Given an undirected graph G , an *orientation* of G is obtained by replacing every edge uv of G by exactly one of the arcs uv and vu . For example the digraph D in Figure 2.1.3 is an orientation of the graph G in Figure 2.1.2. The existence of orientations with certain connectivity properties will be the main subject of Section 3. For now, we wish to give the following basic theorem which is due to Hakimi [58].

Theorem 2.1.1. *Let $G = (V, E)$ be a graph and let $m : V \rightarrow \mathbb{Z}_{\geq 0}$ be a function. Then there is an orientation \vec{G} of G such that $d_{\vec{G}}^-(v) = m(v)$ for all $v \in V$ if and only if $\sum_{v \in V} m(v) = |E|$ and $|E(X)| \leq \sum_{v \in X} m(v)$ for all $X \subseteq V$.*

More generally, an *orientation* of a mixed hypergraph $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ is obtained by replacing every hyperedge $e \in \mathcal{E}$ by a dyperedge a with $\text{tail}(a) \cup \text{head}(a) = e$.

2.1.3 Particular graphs

We here give some important classes of graphs.

2.1.3.1 A list of basic graphs

We first give a list of well-known graphs, first starting with undirected graphs and then switching to directed graphs.

The graph on n vertices that contains exactly one edge between any pair of its vertices is called the *complete* graph \mathbf{K}_n . Given a graph G , we denote the size of the vertex set of a biggest subgraph of G that is a complete graph by $\omega(\mathbf{G})$. We call a connected undirected graph a *path* if it contains exactly two vertices u, v of degree 1 and all other vertices are of degree 2. We call u and v the *endvertices* of the path and speak of a *uv-path*. All other vertices of the path are called *interior* vertices. Given a graph G and $u, v \in V(G)$, we say that v is *reachable* from u if G contains a *uv-path* as a subgraph. The *length* of the path is the number of its edges. We now give a fundamental theorem that relates edge-disjoint paths in a graph to the connectivity properties of a graph. It is due to Menger [93].

Theorem 2.1.2. *Let $G = (V, E)$ be a graph and let $u, v \in V$. Then the maximum number of edge-disjoint paths between u and v in G is exactly $\lambda_G(u, v)$.*

A *cycle* is a 2-regular connected graph. The *length* of the cycle is the number of its edges. A *tree* is a connected graph that does not have a cycle as a subgraph. A vertex that is of degree 1 in a tree is called a *leaf* of the tree.

An illustration can be found in Figure 2.1.5.

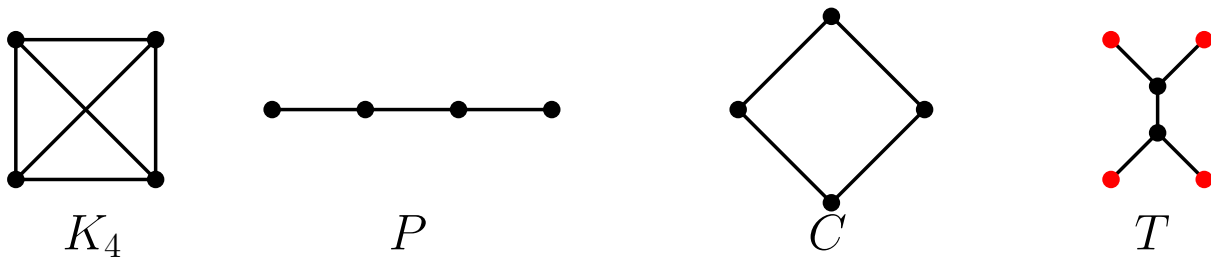


Figure 2.1.5: Examples for the graphs described in Section 2.1.3.1. The graph K_4 is the complete graph on 4 vertices, the graph P is a path of length 3, the graph C is a cycle of length 4 and T is a tree whose leaves are marked in red.

A *hypertree* is a hypergraph that can be trimmed to a tree. We need the following well-known result.

Proposition 2.1.4. *Let $\mathcal{H} = (V, \mathcal{E})$ be a hypertree and $X \subseteq V$. Then there are at most $|X| - 1$ hyperedges which are completely contained in X .*

We now wish to introduce one more graph that has a lot of significant properties and was first mentioned by Kempe [75]. Another such property will be added in Section 3.1.4. The graph is called the *Petersen graph*. It is constructed from two cycles of length 5 $C = v_1, \dots, v_5$ and $C' = w_1, \dots, w_5$ by adding the edges $v_1w_1, v_2w_4, v_3w_2, v_4w_5$ and v_5w_3 . An illustration can be found in Figure 2.1.6.

We now give some similar definitions in directed graphs. An orientation of a complete graph is called a *tournament*. We call a directed graph a *directed path* if its underlying

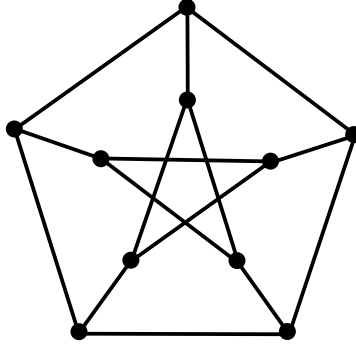


Figure 2.1.6: The Petersen graph.

graph is a path and the in- and out-degree of every vertex is at most 1. Observe that a directed path contains exactly one vertex u of indegree 0 and one vertex v of outdegree 0. We call the directed path a uv -path and say that u and v are the *ends* or *endvertices* of the path. All vertices on the path which are distinct from u and v are called *interior* vertices of the path. Given a digraph D and $u, v \in V(D)$, we say that v is *reachable* from u if D contains a directed uv -path as a subgraph. We now give the directed equivalent of Theorem 2.1.2 which is also due to Menger [93].

Theorem 2.1.3. *Let $D = (V, A)$ be a graph and let $u, v \in V$. Then the maximum number of arc-disjoint directed paths from u to v in D is exactly $\lambda_D(u, v)$.*

We are now ready to give some more notation for mixed hypergraphs.

A *mixed hyperpath* is a mixed hypergraph that can be trimmed to a directed path. If for $u, v \in V$, the directed path can be chosen to go from u to v , we say that the mixed hyperpath is a *mixed hyperpath from u to v* . Given a mixed hypergraph $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ and $u, v \in V$, we say that v is *reachable* from u if there is a mixed hyperpath from u to v .

For a vertex set $X \subseteq V$, we denote by $U_X^{\mathcal{F}}$ the set of vertices which are reachable from the vertices in X by a mixed hyperpath in \mathcal{F} , by $P_X^{\mathcal{F}}$ the set of vertices from which X is reachable by a mixed hyperpath in \mathcal{F} and by $\mathcal{F}[X]$ the mixed subhypergraph of \mathcal{F} induced on X .

A subgraph induced on a maximal set of vertices that are pairwise mutually reachable from each other is called a *strongly connected component* of \mathcal{F} . We now give an analogous result of Proposition 2.1.1 which is also well-known.

Proposition 2.1.5. *There is some $X \subseteq V$ such that $\mathcal{F}[X]$ is a strongly connected component of D and $d_{\mathcal{E}}(X) = d_{\mathcal{A}}^+(X) = 0$.*

In mixed graphs, we abbreviate a mixed hyperpath to a *mixed path*.

Coming back to digraphs, a *circuit* is a strongly connected orientation of a cycle. For both directed paths and circuits, the *length* is the number of arcs. An illustration can be found in Figure 2.1.7.

We will make use of the following simple well-known property of digraphs.

Proposition 2.1.6. *Let $D = (V, A)$ be a digraph with $d_D^-(v) \geq 1$ for all $v \in V$. Then D contains a circuit as a subgraph.*

2.1.3.2 T -joins

In this section, we give a first relation between graph theory and parity. The following well-known result is fundamental.

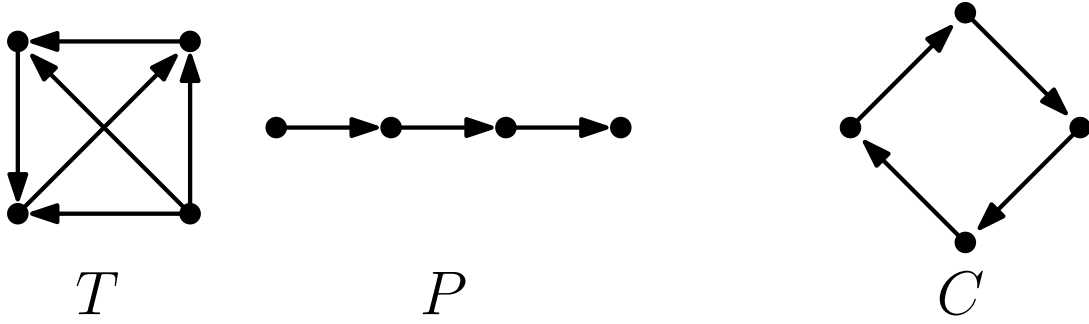


Figure 2.1.7: Examples for the digraphs described in Section 2.1.3.1. The digraph T is a tournament on 4 vertices, the graph P is a directed path of length 3, the graph C is a circuit of length 4.

Proposition 2.1.7. *For any graph $G = (V, E)$, the number of vertices in V which are of odd degree in G is odd.*

Given a graph $G = (V, E)$ and some $T \subseteq V$, a T -join is a set $F \subseteq E$ such that $d_F(v)$ is odd for all $v \in T$ and $d_F(v)$ is even for all $v \in V - T$. While there is a lot of theory on T -joins, we only require the following basic result that can for example be found in [84].

Proposition 2.1.8. *Let $G = (V, E)$ be a connected graph and let $T \subseteq V$. Then G contains a T -join if and only if $|T|$ is even.*

Observe that the necessity in Proposition 2.1.8 follows immediately from Proposition 2.1.7.

2.1.3.3 Eulerian graphs

In this section, we deal with an important graph property. Its definition is slightly different for directed and undirected graphs. We say that an undirected graph is *Eulerian* if the degree of each of its vertices is even. For a directed graph $D = (V, A)$, we say that it is *Eulerian* if $d_D^-(v) = d_D^+(v)$ for all $v \in V$. The following simple results are well-known.

Proposition 2.1.9. *Let $D = (V, A)$ be an Eulerian digraph and $X \subseteq V$. Then $d_D^+(X) = d_D^-(X)$.*

Proposition 2.1.10. *An undirected graph has an Eulerian orientation if and only if it is Eulerian.*

The following result will serve as a motivation in Section 3.2.2. It is an immediate consequence of Proposition 2.1.9.

Proposition 2.1.11. *Every Eulerian orientation of a $2k$ -edge-connected Eulerian graph is k -arc-connected.*

We now give a somewhat stronger statement that will be needed in Section 3.1.4.

Proposition 2.1.12. *Let $G = (V, E)$ be an Eulerian graph and $\{e_v, f_v\}$ two edges incident to v for all $v \in V' \subseteq V$. Then there is an Eulerian orientation of G such that exactly one of e_v and f_v enters v for all $v \in V'$.*

Proof. Let G' be the graph obtained from G by replacing each vertex $v \in V'$ by two vertices u_v and w_v such that in G' , u_v is incident to $\{e_v, f_v\}$ and w_v to $\delta_G(\{v\}) - \{e_v, f_v\}$. As G is Eulerian, so is G' . Hence there exists an Eulerian orientation D' of G' . By contracting u_v and w_v in D' for all $v \in V'$, we obtain the required orientation. \square

The following result can be found in [37] and will be needed in Section 3.1.1.3.

Theorem 2.1.4. *Let G, F be edge-disjoint graphs on the same vertex set V such that $G \cup F$ is an Eulerian graph and let \vec{F} be an orientation of F . Then there is an orientation \vec{G} of G such that $\vec{G} \cup \vec{F}$ is Eulerian if and only if $d_G(X) \geq d_{\vec{F}}^+(X) - d_{\vec{F}}^-(X)$ for all $X \subseteq V$.*

2.1.3.4 Spanning trees

This section deals with an object that plays a significant role in this thesis: spanning trees. Given a graph $G = (V, E)$, a subgraph of G is called *spanning* if it has the same vertex set as G . A spanning subgraph that is a tree is a *spanning tree*. Spanning trees have some interesting properties we work with throughout the thesis.

The following result can be found in a stronger form as Theorem 5.3.3 in [41].

Proposition 2.1.13. *Let $G = (V, E)$ be a graph and let T_1, T_2 be spanning trees of G . Then there is a function $\sigma : E(T_1) \rightarrow E(T_2)$ such that for all $e \in E(T_1)$ both $(T_1 - e) \cup \sigma(e)$ and $(T_2 - \sigma(e)) \cup e$ are spanning trees of G .*

We call a function like in Proposition 2.1.13 a *tree-mapping function* from T_1 to T_2 .

Lemma 2.1.1. *Let $G = (V, E)$ be a graph, let T_1, T_2 be spanning trees of G , and let $\sigma : E(T_1) \rightarrow E(T_2)$ be a tree-mapping function from T_1 to T_2 . Further, let $e_1, e_2, e_3 \in E(T_1)$ be all incident to a common vertex v . Then $\{\sigma(e_1), \sigma(e_2), \sigma(e_3)\}$ contains at least two distinct elements.*

Proof. As T_1 is a spanning tree, $T_1 - \{e_1, e_2, e_3\}$ contains three components C_1, C_2, C_3 none of which contains v such that e_i is incident to a vertex in $V(C_i)$ for $i = 1, 2, 3$. As $(T_1 - e_i) \cup \sigma(e_i)$ is a spanning tree, we obtain that $\sigma(e_i)$ is incident to a vertex in $V(C_i)$ for $i = 1, 2, 3$. As $V(C_1), V(C_2)$ and $V(C_3)$ are pairwise disjoint, the statement follows. \square

We next give a result that is the first of many packing results mentioned in this thesis. It is due to Tutte [106] and Nash-Williams [95].

Theorem 2.1.5. *Let $G = (V, E)$ be a graph and k a positive integer. Then G has k edge-disjoint spanning trees if and only if $e_E(\mathcal{P}) \geq k(q-1)$ for every partition $\mathcal{P} = \{V_1, \dots, V_q\}$ of V .*

The following result is an immediate corollary of Theorem 2.1.5.

Theorem 2.1.6. *Every $2k$ -edge-connected graph has k edge-disjoint spanning trees.*

Further useful properties of spanning trees will be discussed in Section 2.4.

2.1.3.5 Arborescences

We now come to a class of digraphs that is in some way the directed equivalent of spanning trees. An *r -arborescence* is a singularly rooted digraph $B = (V \cup r, A)$ whose underlying graph is a tree and which is root-connected from r . An example can be found in Figure 2.1.8.

We now give a few more notations for an equivalent of arborescences in mixed hypergraphs. A mixed hypergraph \mathcal{F} is called a *mixed hyperarborescence* if all the dyperedges and all the hyperedges of \mathcal{F} can be trimmed to get an arborescence. A *mixed r -hyperarborescence* for some $r \in V$ is a mixed hyperarborescence together with a vertex r where that arborescence can be chosen to be an r -arborescence. A mixed hyperarborescence that does not contain a hyperedge is called a *hyperarborescence*.

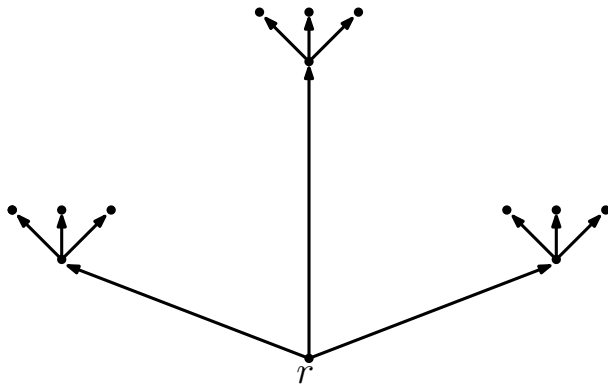


Figure 2.1.8: An r -arborescence.

2.1.3.6 Matchings

Given a graph $G = (V, E)$, a *matching* is a set $M \subseteq E$ such that every $v \in V$ is incident to at most one edge in M . If $d_M(v) = 1$ for all $v \in V$, we say that M is a *perfect matching*. An illustration can be found in Figure 3.2.16.

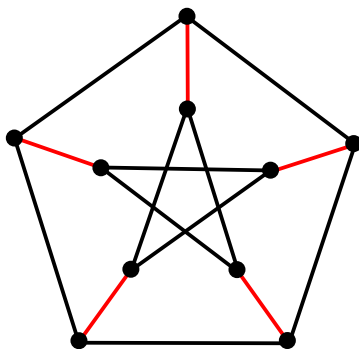


Figure 2.1.9: The edges marked in red form a perfect matching of the Petersen graph.

We need the following theorem that is due to Petersen [99] and guarantees the existence of perfect matchings in a large class of cubic graphs.

Theorem 2.1.7. *Every cubic 2-edge-connected graph has a perfect matching.*

2.2 Tools

In this section, we deal with two important tools that will play an important role in this thesis. The first one is a general concept called submodularity that will be applied in several parts of this thesis. The second one is a well-known operation on graphs called splitting off that will be used in Section 5.

2.2.1 Submodularity

Let S be a set and $f : 2^S \rightarrow \mathbb{R}$ a function. Then f is called *submodular* if $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$ for all $X, Y \subseteq S$. The main motivation for us considering submodular functions comes from the following two connections to graph theory.

Proposition 2.2.1. *Let $G = (V, E)$ be a graph. Then the degree function $d_G : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ is submodular.*

Proposition 2.2.2. *Let $D = (V, A)$ be a digraph. Then the in-degree function $d_D^- : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ is submodular.*

The following consequence of this result will be used in Section 4.4. We say that in a root-connected singularly rooted digraph $D = (V \cup r, A)$, an arc $a \in A$ is *critical* if $D - a$ is not root-connected.

Lemma 2.2.1. *Let $D = (V \cup r, A)$ be a 2-root-connected singularly rooted digraph and let $D' = (V \cup r, A')$ be a root-connected singularly rooted digraph that is obtained from D by deleting α arcs of A . Then for any $v \in V \cup r$, there are at most α arcs which are critical in D' and whose tail is v .*

Proof. We proceed by induction on α . The statement is trivial for $\alpha = 0$. We suppose that it holds for all integers up to some α and show that it also holds for $\alpha + 1$. Let $v \in V \cup r$ and let $\{a_1, \dots, a_{\alpha+1}\}$ be a set of arcs in A such that $D_2 = D - \{a_1, \dots, a_{\alpha+1}\}$ is root-connected. By the inductive hypothesis, v is the tail of at most α critical arcs in $D_1 = D - \{a_1, \dots, a_\alpha\}$. Suppose, for the sake of a contradiction, that there are two arcs vw_1, vw_2 which are critical in D_2 , but not in D_1 . It follows that there are sets $X_1, X_2 \subseteq V$ such that $d_{D_2}^-(X_1), d_{D_2}^-(X_2) = 1$, $d_{D_1}^-(X_1), d_{D_1}^-(X_2) = 2$, vw_1 enters X_1 and vw_2 enters X_2 . Since $D_2 = D_1 - a_{\alpha+1}$, it follows that $a_{\alpha+1}$ enters $X_1 \cap X_2$, so $X_1 \cap X_2 \neq \emptyset$. As D_2 is root-connected, we have $d_{D_2}^-(X_1 \cap X_2) \geq 1$. As vw_i enters X_i , we have $v \in V \cup r - (X_1 \cup X_2)$ and so both vw_1 and vw_2 enter $X_1 \cup X_2$. This yields $d_{D_2}^-(X_1) + d_{D_2}^-(X_2) = 1 + 1 < 1 + 2 \leq d_{D_2}^-(X_1 \cap X_2) + d_{D_2}^-(X_1 \cup X_2)$, a contradiction to Proposition 2.2.2. \square

We further need two more properties that relate the sizes of several cuts in a graph. Both of them are routine and can be verified by counting the contribution of each edge or arc.

Proposition 2.2.3. *Let $G = (V, E)$ be a graph, \vec{G} an orientation of G and $S_1, S_2 \subseteq V$. Then $d_{\vec{G}}^-(S_1 \cup S_2) + d_{\vec{G}}^-(S_1 \cap S_2) = d_{\vec{G}}^-(S_1) + d_{\vec{G}}^-(S_2) - d_G(S_1, S_2)$*

As the following result will later be applied in the capacitated setting, we give them here in the capacitated form.

Proposition 2.2.4. *Let $(G = (V, E), c)$ be a capacitated graph. For all $X, Y \subseteq V$, the following hold:*

- (a) $c(\delta_G(X)) + c(\delta_G(Y)) = c(\delta_G(X \cap Y)) + c(\delta_G(X \cup Y)) + 2c(\delta_G(X, Y))$,
- (b) $c(\delta_G(X)) + c(\delta_G(Y)) = c(\delta_G(X - Y)) + c(\delta_G(X - Y)) + 2c(\bar{\delta}_G(X, Y))$.

2.2.2 Splitting off

We now deal with an operation called splitting off. Given a graph $H = (V \cup s, E)$ with a specified vertex s and two edges $su, sv \in E$ which are incident to s , we define the graph $\mathbf{H}_{u,v}$ by $V(\mathbf{H}_{u,v}) = V \cup s$ and $E(\mathbf{H}_{u,v}) = E - \{su, sv\} \cup uv$. For an illustration, see Figure 2.2.1.

A similar notion exists for directed graphs. Splitting offs were first introduced by Lovász in [87]. We are mainly interested in splitting off in the context of $(2, k)$ -connectivity and therefore postpone all technical results on splitting off to Section 5. Nevertheless, we wish to mention that the theory of maintaining connectivity properties through splitting offs is pretty rich. Two of the most important results were proven by Lovász [87] concerning global connectivity and Mader [89] concerning local connectivity. These results have been useful for both connectivity augmentation problems and constructive characterizations of graphs with certain connectivity properties. For a more detailed survey on the applications of splitting offs, see [41], Chapter 8.

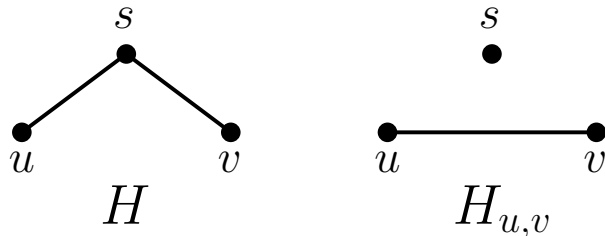


Figure 2.2.1: An illustration of the splitting off operation. The graph $H_{u,v}$ is obtained from H by splitting off the edges su and sv .

2.3 Complexity

In this section, we give the notions on complexity theory we need in this thesis. We first introduce the Landau notation which is the measurement for the running time of the algorithms considered in this thesis and the basis of complexity theory as such. Next, we describe the fundamental complexity classes P and NP and their relationship. Further, we present several NP-complete problems that will serve for reductions in later sections. After, we introduce two complexity classes which provide a finer complexity notion based on the introduction of a parameter. Finally, we introduce a further complexity assumption called the Exponential Time Hypothesis.

2.3.1 Landau notation

We now introduce this classic notation that was developed by Bachmann in the late 19th century in order to evaluate the running time of an algorithm [2]. For an exact definition of an algorithm using Turing machines see [91]. All algorithms considered in this thesis are *deterministic* meaning that they do not take any probabilistic decision. In order to measure the quality of an algorithm, we wish to determine how long it takes to run it for an instance of given size. However, in order to make this measurement easily comparable, we wish to avoid that this running time depends on the concrete implementation of the algorithm or the machine the algorithm is run on. We therefore count the number of basic calculations the algorithm executes for an input of given size where basic calculations mean addition and multiplication of real numbers. However, we are not interested in the absolute number of such calculations but rather in the asymptotic behaviour of this number. More concretely, we say that an algorithm runs in $O(f)$ time for some function f that maps the input size to the integers if there is a constant $c \in \mathbb{R}$ such that for any given input I , the number of basic calculations that the algorithm executes for the input I is at most $cf(|I|)$. Observe that the function f may have several arguments if there are several parameters to measure the size of the input, for example the size of the vertex set and the size of the edge set of a graph. We call an algorithm that runs in $O(f)$ time for some function f an $O(f)$ time algorithm. If $f(n) \leq n^k$ for all $n \in \mathbb{Z}_{\geq 0}$ for some constant $k \in \mathbb{R}$, we say that the algorithm runs in *polynomial time* and call it a *polynomial time algorithm*.

2.3.2 P versus NP

All algorithmic problems considered in this thesis can be viewed as *decision problems*. This means that a problem consists of a characterization of the form of the input and a question related to that input that can be uniquely answered positively or negatively. An *instance* of the problem then is a concrete input for which the problem is to be solved. If the answer is positive (negative) for this instance, we say that the instance is *positive*

(negative). The *complement* of a problem is obtained by exchanging the positive and negative instances of the problem. For most discrete decision problems, trivial but very slow algorithms that decide whether an instance is positive or negative can easily be found. The most interesting question often is whether the problem can be solved by a polynomial time algorithm. In this section, we introduce the two most important complexity classes in this thesis.

2.3.2.1 P

A decision problem is said to be in P if there is an algorithm that correctly solves the problem and runs in polynomial time. Observe that a problem is in P if and only if its complement is in P .

2.3.2.2 NP

We now come to a different complexity class that is seemingly much larger than P . The class NP contains all the problems for which a positive instance can be verified in polynomial time. More precisely, there is a polynomial time algorithm A taking as arguments the instance I and a so-called *witness* w with the following property: For every positive instance I , there is a witness $w(I)$ whose size is polynomial in $|I|$ such that $A(I, w(I))$ is positive and for every negative instance I , we have that $A(I, w)$ is negative for any w . A problem is said to be in $co-NP$ if its complement is in NP .

2.3.2.3 NP-completeness

Observe that every problem in P is also in NP as an empty witness can be used. The other inclusion, known as the $P=NP$ problem, is probably the singularly most important open problem in complexity theory. It has first been considered by Cook [21].

Interestingly, a very concrete approach to try to prove $P=NP$ exists. A problem is called *NP-complete* if it is in NP and the existence of a polynomial time algorithm for the problem implies $P=NP$. A problem is *co-NP-complete* if its complement is NP-complete. Surprisingly, after Cook gave a first NP-complete problem [21], a list of very natural NP-complete problems was provided by Karp [73]. Since then, numerous further NP-complete problems have been found and the notion of NP-completeness has become the most important measure of difficulty for algorithmic problems. Some such problems can be found in Section 2.3.3. We wish to remark that an NP-completeness result also has a structural meaning: if a problem is NP-complete, there is little hope to find a natural characterization of the positive instances. The most common way to prove that a problem P_1 is NP-complete is to find a different problem P_2 which is known to be NP-complete and then show that a polynomial time algorithm for P_1 implies a polynomial time algorithm for P_2 . Such a proof is called a *reduction*.

2.3.3 A collection of NP-complete problems

We here give a collection of NP-complete problems we will need for reductions in the main part of this thesis.

2.3.3.1 3-SAT

We here consider the following problem that was proven to be NP-complete by Karp in [73]. A *binary* variable is a variable that can be assigned exactly one of the values TRUE and FALSE. Given a binary variable x , the two *literals on x* are x and \bar{x} where \bar{x}

denotes the negation of x . For a variable set X , a truth assignment for X is a mapping $\phi : X \rightarrow \{\text{TRUE}, \text{FALSE}\}^X$. Given a truth assignment ϕ for X , we say that a literal $x(\bar{x})$ is *TRUE* if $\phi(x) = \text{TRUE}(\text{FALSE})$. A *clause* is a set of literals.

3-Satisfiability (3-SAT)

Instance A set X of binary variables and a set of clauses \mathcal{C} such that every $C \in \mathcal{C}$ contains 3 literals on X .

Question Is there a truth assignment to the variables of X such that every $C \in \mathcal{C}$ contains at least one true literal?

2.3.3.2 (3,B2)-SAT

The next problem is a more restricted version of 3-SAT. It has been proven that this version remains NP-complete by Berman, Karpinski and Scott in [12].

2-balanced 3-SAT isfiability ((3,B2)-SAT)

Instance A set X of boolean variables, a formula consisting of a set \mathcal{C} of clauses each containing 3 distinct variables with every $x \in X$ appearing exactly twice in positive and twice in negated form.

Question Is there a truth assignment to the variables of X such that every clause in \mathcal{C} contains at least one true literal?

2.3.3.3 MNAE3-SAT and CMNAE3-SAT

We here consider a different variant of the 3-SAT problem.

Monotone Not-all-equal-3-SAT (MNAE3-SAT)

Instance: A set X of boolean variables, a formula consisting of a set \mathcal{C} of clauses each containing 3 distinct literals, none of which are negated.

Question: Is there a truth assignment for the variables of X such that every clause in \mathcal{C} contains at least one true and at least one false literal?

Our interest in this problem is justified by the following result due to Schaefer [101].

Theorem 2.3.1. *MNAE3-SAT is NP-complete.*

Given a MNAE3-SAT formula $F = (X, \mathcal{C})$, we call a truth assignment to the variables of X *feasible* if every clause of \mathcal{C} contains at least one true and at least one false literal. In order to make this problem suitable for a reduction in Section 3.1.4, we need to slightly adapt it. We define the formula graph \mathbf{G}_F by $V(\mathbf{G}_F) = X \cup \mathcal{C}$ and there is an edge between the vertices corresponding to a variable x_i and a clause C_j if x_i is contained in C_j . We call a formula F *connected* if \mathbf{G}_F is connected. We show that MNAE3-SAT stays NP-complete with this additional assumption.

Connected Monotone Not-all-equal-3-SAT (CMNAE3-SAT)

Instance: A set X of boolean variables, a connected formula consisting of a set \mathcal{C} of clauses each containing 3 distinct variables none of which are negated.

Question: Is there a feasible truth assignment for the variables of X ?

Lemma 2.3.1. *CMNAE3-SAT is NP-complete.*

Proof. We show a reduction from MNAE3-SAT. Recall that MNAE3-SAT is NP-complete by Theorem 2.3.1. Let \mathbf{F} be a MNAE3-SAT formula. Let $\mathbf{G}_1, \dots, \mathbf{G}_t$ be the connected components of $G_{\mathbf{F}}$. For $i = 1, \dots, t$, consider the MNAE3-SAT formula \mathbf{F}_i that consists of the variables and clauses corresponding to vertices in G_i . Observe that $G_{\mathbf{F}_i} = G_i$ and so every \mathbf{F}_i is an instance of CMNAE3-SAT. We will show that \mathbf{F} is a positive instance of MNAE3-SAT if and only if all of the \mathbf{F}_i are positive instances of CMNAE3-SAT. First assume that there is a feasible truth assignment for \mathbf{F} . The restriction of this assignment to the variables of \mathbf{F}_i yields a feasible truth assignment for \mathbf{F}_i for all $i = 1, \dots, t$. Now assume that there is a feasible truth assignment for \mathbf{F}_i for $i = 1, \dots, t$. As every vertex corresponding to a variable is contained in exactly one component, every variable is contained in exactly one of the \mathbf{F}_i and so we obtain a unique assignment of boolean values to all variables. As every clause of \mathcal{C} is contained in some \mathbf{F}_i , this assignment is feasible for \mathbf{F} . This finishes the proof. \square

2.3.3.4 MAXCUT and AMAXCUT

We introduce one more NP-complete problem that is of slightly different nature. The unweighted MAXCUT problem can be formulated as follows:

MAXCUT:

Instance: A graph $H = (V, E)$ and a positive integer k .

Question: Is there some $X \subseteq V$ such that $d_H(X) > k$?

A proof of the following theorem can be found in [56].

Theorem 2.3.2. *MAXCUT is NP-complete.*

For a reduction in Section 3.1.1, we need a slightly adapted version of MAXCUT.

Adapted MAXCUT (**AMAXCUT**):

Instance: An Eulerian graph $H = (V, E)$ such that $|E| \geq 6$ is even and an even positive integer k .

Question: Is there some $X \subseteq V$ such that $d_H(X) > k$?

Lemma 2.3.2. *AMAXCUT is NP-complete.*

Proof. We show this by a reduction from MAXCUT. Let $(\mathbf{H} = (\mathbf{V}, \mathbf{E}), \mathbf{k})$ be an instance of MAXCUT. We may obviously suppose that $|\mathbf{E}| \geq 3$. Let $\mathbf{H}' = (\mathbf{V}, \mathbf{E}')$ be the graph which is obtained from \mathbf{H} by replacing every edge of \mathbf{E} by 2 parallel copies of itself. Observe that $|\mathbf{E}'| = 2|\mathbf{E}| \geq 6$ is even and $d_{\mathbf{H}'}(v) = 2d_{\mathbf{H}}(v)$ is even for all $v \in \mathbf{V}$. Further, for every $X \subseteq \mathbf{V}$, we have $d_{\mathbf{H}'}(X) = 2d_{\mathbf{H}}(X)$. This yields that (\mathbf{H}, \mathbf{k}) is a positive instance of MAXCUT if and only if $(\mathbf{H}', 2\mathbf{k})$ is a positive instance of AMAXCUT. \square

2.3.4 XP and FPT

We now introduce two more complexity classes. For both of these complexity classes, a parameter k is part of the input. The parameter k can either be explicitly specified in the input or it can be implicitly contained in the remaining input, for example it can be an invariant of a graph that is part of the input. We now say that a problem is in *XP* with respect to the parameter k if there is a function $c : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ and an algorithm that correctly solves the problem and that for any given input I with parameter k runs in $O(|I|^{c(k)})$. Further, we say that a problem is in *FPT* with respect to the parameter k if there is a function $f : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ and an algorithm that correctly solves the problem and that for any given input I with parameter k runs in $f(k)|I|^{O(1)}$. For a survey on parameterized complexity, see [22].

2.3.5 The Exponential Time Hypothesis (ETH)

We now come to a further complexity assumption that was introduced by Impagliazzo and Paturi in [67]. While it is somewhat weaker than the assumption that $P \neq NP$, it is also considered a rather mild assumption in complexity theory. The *Exponential Time Hypothesis (ETH)* states that there is some $\epsilon > 0$ such that there is no algorithm that solves 3-SAT correctly and runs in $2^{\epsilon \ell} \cdot (\ell + m)^{O(1)}$ where m denotes the number of variables and ℓ denotes the number of clauses of the formula. We further need the following result that shows that the ETH is equivalent to a seemingly slightly weaker version of itself.

Lemma 2.3.3 (Impagliazzo et al. [68]). *Assuming the ETH, there is some $\epsilon > 0$ such that there is no algorithm that solves 3-SAT correctly and runs in $2^{\epsilon m} \cdot (\ell + m)^{O(1)}$ where m denotes the number of variables and ℓ denotes the number of clauses of the formula.*

2.4 Matroids

We now deal with an object called matroids that was introduced by Whitney in 1935 in [111]. For a deep survey on the theory of matroids, see [98]. For us, matroids will mainly serve as a tool in Section 4. We now proceed to the formal definition of a matroid. Note that there are several equivalent ways to define a matroid. We choose the one that is most convenient for our purposes.

Definition 2.4.1. *A matroid is a set S called the ground set together with a collection $\mathcal{I} \subseteq 2^S$ called the collection of independent sets that satisfies the following properties:*

$$(I_0) \quad \emptyset \in \mathcal{I},$$

$$(I_1) \quad \text{if } Z \subsetneq Z' \text{ and } Z' \in \mathcal{I}, \text{ then } Z \in \mathcal{I},$$

$$(I_2) \quad \text{if } Z, Z' \in \mathcal{I} \text{ and } |Z| < |Z'|, \text{ then there exists some } x \in Z' - Z \text{ such that } Z \cup \{x\} \in \mathcal{I}.$$

For a matroid M , we denote by $\mathcal{I}(M)$ the collection of *independent* sets of M . A set $X \subseteq S$ that is not independent is called *dependent*. A maximal independent set of M is called a *basis* of M . We next define a function $r_M: 2^S \rightarrow \mathbb{Z}_{\geq 0}$, called the *rank function* of M by $r_M(Z) = \max\{|Y| : Y \subseteq Z, Y \in \mathcal{I}(M)\}$.

The following routine result can be found in a similar form as Theorem 5.3.7 in [41].

Proposition 2.4.1. *Let M be a matroid with ground set S . Then r_M satisfies the following properties:*

$$(R_0) \quad 0 \leq r_M(Z) \leq |Z| \text{ for every } Z \subseteq S,$$

$$(R_1) \quad r_M(Y) \leq r_M(Z) \text{ for all } Y, Z \subseteq S \text{ with } Y \subseteq Z,$$

$$(R_2) \quad r_M \text{ is submodular.}$$

Further, every integer valued function satisfying $(R_0) - (R_2)$ is the rank function of a unique matroid on S .

It hence follows that a matroid is uniquely determined by its ground set and rank function. In the following, we often define a matroid by $M = (\mathbf{S}, \mathbf{r})$ meaning that S is the ground set of the matroid and r is its rank function. We use $\mathbf{r}(M)$ for $r(S)$. The following basic result is well-known.

Proposition 2.4.2. *Let $M = (S, r)$ be a matroid and let B be a basis of M . Then $|B| = r(S)$.*

Further, we define $\mathbf{span}_M(\mathbf{Z}) = \{s \in S : r_M(Z \cup s) = r_M(Z)\}$. We need the following simple, well-known property.

Proposition 2.4.3. *Let $M = (S, r)$ be a matroid and $X_1, \dots, X_t \subseteq S$ for some $t \in \mathbb{Z}_{\geq 0}$. Then $\bigcup_{i \in \{1, \dots, t\}} \mathbf{span}_M(X_i) \subseteq \mathbf{span}_M(\bigcup_{i \in \{1, \dots, t\}} X_i)$.*

Some $s \in S$ is called a *loop* in M if $\{s\}$ is dependent in M . Two elements $s_1, s_2 \in S$ are called *parallel* in M if none of s_1, s_2 is a loop and $\{s_1, s_2\}$ is dependent in M .

2.4.1 General results on matroids

We here describe three basic results on matroids. Firstly, we show that for two matroids on a given ground set, the structure of sets which are independent in both matroids is well understood. Next, we give an operation that allows to add an element to a matroid. After, we give two operations that can be applied to obtain a new matroid from two other ones.

2.4.1.1 Matroid intersection

We here describe two theorems that deal with the problem of computing a common independent set of fixed size of two given matroids on the same ground set. The first one is of structural nature and was proven by Edmonds in [30].

Theorem 2.4.1. *Let $M_1 = (S, r_1)$ and $M_2 = (S, r_2)$ be two matroids on a common ground set S and μ a positive integer. Then a common independent set of size μ of M_1 and M_2 exists if and only if $r_1(Z) + r_2(S - Z) \geq \mu$ for all $Z \subseteq S$.*

The next theorem is of algorithmic nature. It is also due to Edmonds [29]. Given a matroid M on ground set S , an algorithm that decides whether a given subset of S is independent in M is called an *independence oracle* for M .

Theorem 2.4.2. *Let $M_1 = (S, r_1)$ and $M_2 = (S, r_2)$ be two matroids on a common ground set S with polynomial independence oracles for M_1 and M_2 being available, μ a positive integer and $w : S \rightarrow \mathbb{R}$ a weight function. We can decide if a common independent set of size μ of M_1 and M_2 exists in polynomial time. Further, if this is the case, then one of minimum weight can be computed in polynomial time.*

2.4.1.2 Parallel extension

We now describe a simple well-known method to add an element to a matroid.

Lemma 2.4.1. *Let $M = (S, r_M)$ be a matroid, $s \in S$ and t a new element that is not in S . Then there is a unique matroid $M' = (S \cup t, r_{M'})$ on $S \cup t$ such that $r_{M'}(Z) = r_M(Z)$ and $r_{M'}(Z \cup \{s, t\}) = r_{M'}(Z \cup t) = r_{M'}(Z \cup s) = r_M(Z \cup s)$ for all $Z \subseteq S - s$.*

Given a matroid $M = (S, r_M)$, some $s \in S$ and some integer $k \geq 1$, we say that we *replace s by k parallel copies of itself* if we replace M by the matroid that is obtained applying the above operation $k - 1$ times for the element s .

2.4.1.3 Direct matroid sum

We now introduce a simple matroid operation that merges two matroids into a bigger one. It is also described in Section 5.2.2 of [41].

Theorem 2.4.3. *Let M_1, M_2 be matroids on disjoint ground sets S_1, S_2 . Then $\{I_1 \cup I_2 \subseteq S_1 \cup S_2 : I_1 \in \mathcal{I}(M_1), I_2 \in \mathcal{I}(M_2)\}$ forms the collection of independent sets of a matroid, called the direct sum of M_1 and M_2 . Moreover, if a polynomial time independence oracle for both M_1 and M_2 is available, then a polynomial time independence oracle for the direct sum of M_1 and M_2 is also available.*

2.4.1.4 Matroid sum

We now discuss one more well-known operation on matroids. Even though its definition has some similarity with direct matroid sums, the difference is significant. The following result was first proven by Edmonds and Fulkerson in [32].

Theorem 2.4.4. *Let M_1, M_2 be matroids on a common ground set S . Then $\{I_1 \cup I_2 \subseteq S : I_1 \in \mathcal{I}(M_1), I_2 \in \mathcal{I}(M_2)\}$ forms the collection of independent sets of a matroid, called the sum of M_1 and M_2 . Moreover, if a polynomial time independence oracle for both M_1 and M_2 is available, then a polynomial time independence oracle for the sum of M_1 and M_2 is also available.*

By repeatedly applying Theorem 2.4.4, we obtain the following result that will be used several times in Section 2.4.2.2.

Theorem 2.4.5. *Let M be a matroid on a ground set S and k a positive integer. Then $\{I_1 \cup \dots \cup I_k \subseteq S : I_j \in \mathcal{I}(M) \text{ for } j = 1, \dots, k\}$ forms the collection of independent sets of a matroid we denote by \mathbf{M}^k . Moreover, if a polynomial time independence oracle for M is available, then a polynomial time independence oracle for \mathbf{M}^k is also available.*

2.4.2 Particular matroids

We here introduce a collection of matroids that will prove useful in Section 4.

2.4.2.1 matroids and generalized partition matroids

Given a ground set S of size n and some nonnegative integer $k \leq n$, it is easy to see that $\{Z \subseteq S : |Z| \leq k\}$ forms the collection of independent sets of a matroid. This matroid is called the *uniform matroid* $\mathbf{U}_{n,k}$. The uniform matroid $\mathbf{U}_{n,n}$ is called the *free* matroid on the set S . A matroid that is obtained as the direct sum of several uniform matroids is called a *partition matroid*.

We now describe a partition matroid that arises in the context of arborescence packings. It has been introduced in a slightly more stricted setting by Edmonds [31]. Given a simply rooted dypergraph $\mathcal{D} = (V \cup R, \mathcal{A})$, its *k -entering matroid* is the direct sum of the uniform matroids of rank k on $\delta_{\mathcal{D}}^-(v)$ for all $v \in V$. Observe that the ground set of this matroid is \mathcal{A} as R is a set of roots.

While partition matroids are well studied, we now come to a less common, much more general matroid which was considered in a similar form by Frank in [41], see Problem 5.3.4.

Let $\{\mathcal{S}_1, \dots, \mathcal{S}_n\}$ be a partition of a set \mathcal{S} and $\mu, \alpha_i, \beta_i \in \mathbb{Z}$ for all $i \in \{1, \dots, n\}$. For $Z \subseteq \mathcal{S}$ and $i \in \{1, \dots, n\}$, let $z_i = |Z \cap \mathcal{S}_i|$. Let

$$\mathcal{I} = \{Z \subseteq \mathcal{S} : z_i \leq \beta_i \text{ for all } i \in \{1, \dots, n\}, \sum_{i=1}^n \max\{\alpha_i, z_i\} \leq \mu\},$$

$$\mathcal{B} = \{Z \subseteq \mathcal{S} : \alpha_i \leq z_i \leq \beta_i \text{ for all } i \in \{1, \dots, n\}, |Z| = \mu\} \text{ and}$$

$$\mathbf{r}(Z) = \min\{\sum_{i=1}^n \min\{\beta_i, z_i\}, \mu - \sum_{i=1}^n \max\{\alpha_i - z_i, 0\}\} \text{ for all } Z \subseteq \mathcal{S}.$$

We acknowledge that there is a different proof of the following result due to Szegedi [103]. It is shorter but it uses rather sophisticated methods of the theory of generalized polymatroids.

Theorem 2.4.6. *There exists a matroid M whose set of independent sets is \mathcal{I} , set of bases is \mathcal{B} and rank function is r if and only if*

$$\max\{\alpha_i, 0\} \leq \min\{\beta_i, |S_i|\} \text{ for all } i \in \{1, \dots, n\}, \quad (2.4.1)$$

$$\sum_{i=1}^n \max\{\alpha_i, 0\} \leq \mu \leq \sum_{i=1}^n \min\{\beta_i, |S_i|\}. \quad (2.4.2)$$

The matroid M in Theorem 2.4.6 is called *generalized partition matroid*.

Proof. First suppose that M is a matroid and let $Z \in \mathcal{B}$. For all $i \in \{1, \dots, n\}$, this yields $\max\{\alpha_i, 0\} \leq z_i \leq \min\{\beta_i, |S_i|\}$. Further, we obtain $\sum_{i=1}^n \max\{\alpha_i, 0\} \leq \sum_{i=1}^n z_i = \mu$ and $\sum_{i=1}^n \max\{\beta_i, |S_i|\} \geq \sum_{i=1}^n z_i = \mu$.

We now show sufficiency through three claims.

Claim 2.4.1. \mathcal{I} forms the collection of independent sets of a matroid M .

Proof. We need to prove that \mathcal{I} satisfies $(I_0) - (I_2)$ from Definition 2.4.1.

(I_0) : By (2.4.1), we have $\beta_i \geq 0 = |\emptyset \cap S_i|$ for $i \in \{1, \dots, n\}$. Further, (2.4.2) yields $\sum_{i=1}^n \max\{\alpha_i, |\emptyset \cap S_i|\} = \sum_{i=1}^n \max\{\alpha_i, 0\} \leq \mu$. This yields $\emptyset \in \mathcal{I}$.

(I_1) : Let $Z \subsetneq Z' \in \mathcal{I}$. Then $z_i \leq z'_i \leq \beta_i$ for $i = 1, \dots, n$ and $\sum_{i=1}^n \max\{\alpha_i, z_i\} \leq \sum_{i=1}^n \max\{\alpha_i, z'_i\} \leq \mu$, so we have $Z \in \mathcal{I}$.

(I_2) : Let $Z, Z' \in \mathcal{I}$ and $|Z| < |Z'|$. Let $J = \{j \in \{1, \dots, n\} : z_j < z'_j\}$. Observe that $J \neq \emptyset$ as $|Z| < |Z'|$. For all $j \in J$, let $x_j \in (S_j \cap Z') - Z$ and $Z^j = Z \cup \{x_j\}$. Observe that for all $j \in J$, we have $z_j^j \leq z'_j \leq \beta_j$ and $z_i^j \leq z_i \leq \beta_i$ for all $i \in \{1, \dots, n\} - \{j\}$. In order to prove that $Z^j \in \mathcal{I}$ for some $j \in J$, it remains to show that there is some $j \in J$ with $\sum_{i=1}^n \max\{\alpha_i, z_i^j\} \leq \mu$. If there is some $j \in J$ with $z_j < \alpha_j$, then $\sum_{i=1}^n \max\{\alpha_i, z_i^j\} = \sum_{i=1}^n \max\{\alpha_i, z_i\} \leq \mu$, so we are done. We may hence suppose that $z_j \geq \alpha_j$ for all $j \in J$. This yields $\max\{\alpha_j - z'_j, 0\} \geq 0 = \max\{\alpha_j - z_j, 0\}$ for all $j \in J$. For all $i \in \{1, \dots, n\} - J$, we have $z_i \geq z'_i$ yielding $\max\{\alpha_i - z'_i, 0\} \geq \max\{\alpha_i - z_i, 0\}$. This yields $\max\{\alpha_i - z'_i, 0\} \geq \max\{\alpha_i - z_i, 0\}$ for all $i \in \{1, \dots, n\}$. As $|Z| < |Z'|$ and $Z' \in \mathcal{I}$, for some arbitrary $j \in J$ we obtain

$$\begin{aligned} \sum_{i=1}^n \max\{\alpha_i, z_i^j\} &\leq \sum_{i=1}^n \max\{\alpha_i, z_i\} + 1 \\ &= \sum_{i=1}^n \max\{\alpha_i - z_i, 0\} + \sum_{i=1}^n z_i + 1 \\ &\leq \sum_{i=1}^n \max\{\alpha_i - z'_i, 0\} + \sum_{i=1}^n z'_i \\ &= \sum_{i=1}^n \max\{\alpha_i, z'_i\} \\ &\leq \mu. \end{aligned}$$

□

Claim 2.4.2. $Z \in \mathcal{B}$ if and only if Z is a maximal element in \mathcal{I} .

Proof. First let $Z \in \mathcal{B}$. We obtain $\mu = |Z| = \sum_{i=1}^n z_i = \sum_{i=1}^n \max\{\alpha_i, z_i\}$. As $\beta_i \geq z_i$ for all $i \in \{1, \dots, n\}$, we have $Z \in \mathcal{I}$. Further, for any proper superset Z' of Z , we have $\sum_{i=1}^n \max\{\alpha_i, z'_i\} \geq |Z'| > |Z| = \mu$, so $Z' \notin \mathcal{I}$. It follows that Z is maximally in \mathcal{I} .

Now let Z be a maximal element in \mathcal{I} . If $z_j < \alpha_j$ for some $j \in \{1, \dots, n\}$, let $x \in S_j - Z$ and let $Z' = Z \cup \{x\}$. As $\alpha_j \leq \beta_j$ and $Z \in \mathcal{I}$, we obtain $z'_j \leq z_j + 1 \leq \alpha_j \leq \beta_j$ and $z'_i = z_i \leq \beta_i$ for all $i \in \{1, \dots, n\} - \{j\}$. Further, $\sum_{i=1}^n \max\{\alpha_i, z'_i\} = \sum_{i=1}^n \max\{\alpha_i, z_i\} \leq \mu$, so $Z' \in \mathcal{I}$. This contradicts the maximality of Z . We obtain that $z_j \geq \alpha_j$ for all $j \in \{1, \dots, n\}$.

If $|Z| < \mu$, by $\sum_{i=1}^n \min\{\beta_i, |S_i|\} \geq \mu$, there exists some $j \in \{1, \dots, n\}$ such that $z_j < \min\{\beta_j, |S_j|\}$. Let $x \in S_j - Z$ and let $Z' = Z \cup \{x\}$. We have $z'_j \leq z_j + 1 \leq \beta_j$ and $z'_i = z_i \leq \beta_i$ for all $i \in \{1, \dots, n\} - \{j\}$. Further, $\sum_{i=1}^n \max\{\alpha_i, z'_i\} = \sum_{i=1}^n \max\{\alpha_i, z_i\} + 1 = |Z| + 1 \leq \mu$, so $Z' \in \mathcal{I}$. This contradicts the maximality of Z . It follows that $Z \in \mathcal{B}$. \square

Claim 2.4.3. *The rank function of M is r .*

Proof. Let $Z \subseteq S$ and Y be a maximal element of \mathcal{I} in Z .

As $Y \in \mathcal{I}$ and $Y \subseteq Z$, we obtain $y_i \leq \min\{\beta_i, z_i\}$ for all $i \in \{1, \dots, n\}$. This yields $r_M(Z) = |Y| \leq \sum_{i=1}^n \min\{\beta_i, z_i\}$. Further, as $y_i \leq z_i$ for all $i \in \{1, \dots, n\}$ and $Y \in \mathcal{I}$, we obtain $\sum_{i=1}^n y_i + \sum_{i=1}^n \max\{\alpha_i - z_i, 0\} = \sum_{i=1}^n \max\{\alpha_i - z_i + y_i, y_i\} \leq \sum_{i=1}^n \max\{\alpha_i, y_i\} \leq \mu$, so $r_M(Z) = |Y| \leq \mu - \sum_{i=1}^n \max\{\alpha_i - z_i, 0\}$. It follows that $r_M(Z) \leq \min\{\sum_{i=1}^n \min\{\beta_i, z_i\}, \mu - \sum_{i=1}^n \max\{\alpha_i - z_i, 0\}\} = r(Z)$.

Let $J = \{j \in \{1, \dots, n\} : y_j < \min\{z_j, \beta_j\}\}$. If $J = \emptyset$, we obtain $r_M(Z) = |Y| \geq \sum_{i=1}^n \min\{z_i, \beta_i\} \geq \min\{\sum_{i=1}^n \min\{\beta_i, z_i\}, \mu - \sum_{i=1}^n \max\{\alpha_i - z_i, 0\}\} = r(Z)$, so we are done. We may hence suppose that $J \neq \emptyset$.

For all $j \in J$, let $x_j \in (S_j \cap Z) - Y$ and let $Y^j = Y \cup \{x_j\}$. Observe that for all $j \in J$, we have $y_j^j = y_j + 1 \leq \min\{z_j, \beta_j\}$ and $y_i^j = y_i \leq \min\{z_i, \beta_i\}$ for all $i \in \{1, \dots, n\} - \{j\}$. If $\sum_{i=1}^n \max\{\alpha_i, y_i\} < \mu$, then for some arbitrary $j \in J$, we have $\sum_{i=1}^n \max\{\alpha_i, y_i^j\} \leq \sum_{i=1}^n \max\{\alpha_i, y_i\} + 1 \leq \mu$, so $Y^j \in \mathcal{I}$, a contradiction to the maximality of Y . This yields $\sum_{i=1}^n \max\{\alpha_i, y_i\} = \mu$.

If there is some $j \in J$ such that $y_j < \alpha_j$, then $\sum_{i=1}^n \max\{\alpha_i, y_i^j\} = \sum_{i=1}^n \max\{\alpha_i, y_i\} = \mu$, so $Y^j \in \mathcal{I}$, a contradiction to the maximality of Y . This yields $y_j \geq \alpha_j$ for all $j \in J$, so, as $y_j \leq z_j$, we obtain $\max\{\alpha_j - y_j, 0\} = 0 = \max\{\alpha_j - z_j, 0\}$. For all $i \in \{1, \dots, n\} - J$, we have either $y_i = \beta_i$ or $y_i = z_i$. If $y_i = \beta_i$, by $y_i \leq z_i$ and $\alpha_i \leq \beta_i$, we obtain $\max\{\alpha_i - y_i, 0\} = 0 = \max\{\alpha_i - z_i, 0\}$. If $y_i = z_i$, we clearly obtain $\max\{\alpha_i - y_i, 0\} = \max\{\alpha_i - z_i, 0\}$. It follows that $\max\{\alpha_i - y_i, 0\} = \max\{\alpha_i - z_i, 0\}$ holds for all $i \in \{1, \dots, n\}$.

This yields $r_M(Z) = \sum_{i=1}^n y_i = \sum_{i=1}^n \max\{\alpha_i, y_i\} - \sum_{i=1}^n \max\{\alpha_i - y_i, 0\} = \mu - \sum_{i=1}^n \max\{\alpha_i - z_i, 0\} \geq \min\{\sum_{i=1}^n \min\{\beta_i, z_i\}, \mu - \sum_{i=1}^n \max\{\alpha_i - z_i, 0\}\} = r(Z)$. \square

The three previous claims yield the theorem. \square

We are now ready to apply this result to a matroid that generalizes the k -entering matroid of a dypergraph.

Corollary 2.4.1. *Let $\mathcal{D} = (V, \mathcal{A}')$ be a directed hypergraph and $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$ integer functions such that the following two conditions are satisfied:*

$$\max\{k - g(v), 0\} \leq \min\{k - f(v), d_{\mathcal{A}'}^-(v)\} \text{ for all } v \in V, \quad (2.4.3)$$

$$\sum_{v \in V} \max\{k - g(v), 0\} \leq k(|V| - 1) \leq \sum_{v \in V} \min\{k - f(v), d_{\mathcal{A}'}^-(v)\}. \quad (2.4.4)$$

Then $\{\mathcal{Z} \subseteq \mathcal{A}' : |\mathcal{Z}| = k(|V| - 1), f(v) \leq k - d_{\mathcal{Z}}^-(v) \leq g(v) \text{ for all } v \in V\}$ is the set of bases of a matroid $M_{\mathcal{D}}^{(k,f,g)}$ on \mathcal{A}' with rank function

$$r_{M_{\mathcal{D}}^{(k,f,g)}}(\mathcal{Z}) = \min\left\{\sum_{v \in V} \min\{k - f(v), d_{\mathcal{Z}}^-(v)\}, k(|V| - 1) - \sum_{v \in V} \max\{k - g(v) - d_{\mathcal{Z}}^-(v), 0\}\right\}.$$

Proof. Let $\mathcal{S} = \mathcal{A}'$, $\mu = k(|V| - 1)$ and $\mathcal{S}_v = \delta_{\mathcal{A}'}^-(v)$, $\alpha_v = k - g(v)$ and $\beta_v = k - f(v)$ for all $v \in V$. Then (2.4.3) and (2.4.4) coincide with (2.4.1) and (2.4.2), respectively. We can therefore apply Theorem 2.4.6 from which the statement immediately follows. \square

The following is an immediate corollary of the definition of $M_{\mathcal{D}}^{(k,f,g)}$.

Lemma 2.4.2. *Given a directed hypergraph $\mathcal{D} = (V, \mathcal{A}')$, $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$ integer functions satisfying (2.4.3) and (2.4.4) and $\mathcal{Z} \subseteq \mathcal{A}'$, we can decide in polynomial time if \mathcal{Z} is independent in $M_{\mathcal{D}}^{(k,f,g)}$.*

2.4.2.2 Hypergraphic matroids

We here give a construction of a matroid associated to a hypergraph. Interestingly, while the construction is well-known when restricted to graphs, its generalization to hypergraphs has received significantly less attention. The following matroid construction was first observed by Lorea [86] and was later rediscovered by Frank, Király and Kriesell [51]. Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$, let $\mathcal{I}_{\mathcal{H}} = \{\mathcal{Z} \subseteq \mathcal{E} : |V(\mathcal{Z}')| > |\mathcal{Z}'| \text{ for all } \emptyset \neq \mathcal{Z}' \subseteq \mathcal{Z}\}$.

Theorem 2.4.7. *The set $\mathcal{I}_{\mathcal{H}}$ is the set of independent sets of a matroid $M_{\mathcal{H}}$ on \mathcal{E} .*

The matroid $M_{\mathcal{H}}$ is called the *hypergraphic matroid* of the hypergraph \mathcal{H} . If \mathcal{H} is a graph, we call $M_{\mathcal{H}}$ the *graphic matroid* of \mathcal{H} .

For an algorithmic result in Section 4.3, we need to show that an independence oracle for $M_{\mathcal{H}}$ exists. In order to do so, we require the following two preliminaries. The first result can be found as Corollary 2.6 in [51].

Proposition 2.4.4. *Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph. Then $r_{M_{\mathcal{H}}}(\mathcal{E}) = |V| - 1$ if and only if \mathcal{H} is partition-connected.*

This result is useful due to the next one which can be found in [41] as a comment after Theorem 9.1.22 stating that the proof of Theorem 9.1.15 is algorithmic.

Proposition 2.4.5. *There is a polynomial time algorithm that decides whether a given hypergraph \mathcal{H} is partition-connected.*

We are now ready to conclude that a polynomial time independence oracle for $M_{\mathcal{H}}$ exists.

Lemma 2.4.3. *Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$ and $\mathcal{Z} \subseteq \mathcal{E}$, we can decide in polynomial time whether \mathcal{Z} is independent in $M_{\mathcal{H}}$.*

Proof. If $|\mathcal{Z}| \geq |V|$, it follows immediately from the definition of $M_{\mathcal{H}}$ that \mathcal{Z} is dependent in $M_{\mathcal{H}}$. We may hence suppose that $|\mathcal{Z}| \leq |V| - 1$. Let a hypergraph \mathcal{H}' be obtained from (V, \mathcal{Z}) by adding a set \mathcal{S} of $|V| - 1 - |\mathcal{Z}|$ hyperedges each of which equals V . Observe that $|\mathcal{Z} \cup \mathcal{S}| = |V| - 1$.

Claim 2.4.4. *\mathcal{Z} is independent in $M_{\mathcal{H}}$ if and only if \mathcal{H}' is partition-connected.*

Proof. First suppose that \mathcal{Z} is independent in $M_{\mathcal{H}}$. By definition of $M_{\mathcal{H}}$, for any $\mathcal{Z}' \subseteq \mathcal{Z}$, we have $|V(\mathcal{Z}')| > |\mathcal{Z}'|$. For any $\mathcal{Z}' \subseteq \mathcal{Z} \cup \mathcal{S}$ with $\mathcal{Z}' \cap \mathcal{S} \neq \emptyset$, we have $|V(\mathcal{Z}')| = |V| > |\mathcal{Z} \cup \mathcal{S}| \geq |\mathcal{Z}'|$. It follows that $|V(\mathcal{Z}')| > |\mathcal{Z}'|$ for all $\mathcal{Z}' \subseteq \mathcal{Z} \cup \mathcal{S}$ and so by definition, $\mathcal{Z} \cup \mathcal{S}$ is independent in $M_{\mathcal{H}'}$, so $|\mathcal{Z} \cup \mathcal{S}| = r_{M_{\mathcal{H}'}}(\mathcal{Z} \cup \mathcal{S}) = |V| - 1$. Now Proposition 2.4.4 yields that \mathcal{H}' is partition-connected.

Now suppose that \mathcal{H}' is partition-connected. It follows from Proposition 2.4.4 that $r_{M_{\mathcal{H}'}}(\mathcal{Z} \cup \mathcal{S}) = |V| - 1 = |\mathcal{Z} \cup \mathcal{S}|$. It follows that $M_{\mathcal{H}'}$ is the free matroid, so in particular, \mathcal{Z} is independent in $M_{\mathcal{H}'}$. It follows that \mathcal{Z} is also independent in $M_{\mathcal{H}}$. \square

By Claim 2.4.4 and as \mathcal{H}' can be constructed efficiently, it suffices to check whether \mathcal{H}' is partition-connected. By Proposition 2.4.5, this can be done in polynomial time. \square

We also need the k -sum matroid of $M_{\mathcal{H}}$, that is the matroid on ground set \mathcal{E} in which a subset of \mathcal{E} is independent if it can be partitioned into k independent sets of $M_{\mathcal{H}}$. We call this matroid k -hypergraphic matroid and refer to it as $\mathbf{M}_{\mathcal{H}}^k$. The following formula for the rank function of $M_{\mathcal{H}}^k$ was proved by Frank, Király and Kriesell [51].

Theorem 2.4.8. *For all $\mathcal{Z} \subseteq \mathcal{E}$, we have $r_{M_{\mathcal{H}}^k}(\mathcal{Z}) = \min\{e_{\mathcal{Z}}(\mathcal{P}) + k(|V| - |\mathcal{P}|) : \mathcal{P} \text{ a partition of } V\}$.*

We now extend the previous construction to mixed hypergraphs. Let $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ be a mixed hypergraph, $\mathcal{H}_{\mathcal{F}} = (V, \mathcal{E}_{\mathcal{A}} \cup \mathcal{E})$ the underlying hypergraph of \mathcal{F} and $\mathcal{D}_{\mathcal{F}} = (V, \mathcal{A} \cup \mathcal{A}_{\mathcal{E}})$ the directed extension of \mathcal{F} . We now construct the *extended k -hypergraphic matroid* $\mathbf{M}_{\mathcal{F}}^k$ on $\mathcal{A} \cup \mathcal{A}_{\mathcal{E}}$ from $M_{\mathcal{H}_{\mathcal{F}}}^k$ by replacing every $e \in \mathcal{E}$ by $|e|$ parallel copies of itself, associating these elements to the dyperedges in \mathcal{A}_e and associating every element of $\mathcal{E}_{\mathcal{A}}$ to the corresponding element in \mathcal{A} . We give the following formula for the rank function of $M_{\mathcal{F}}^k$.

Proposition 2.4.6. *For all $\mathcal{Z} \subseteq \mathcal{A} \cup \mathcal{A}_{\mathcal{E}}$, we have*

$$r_{M_{\mathcal{F}}^k}(\mathcal{Z}) = \min\{|\mathcal{Z} \cap \mathcal{A}(\mathcal{P})| + |\{e \in \mathcal{E}(\mathcal{P}) : \mathcal{Z} \cap \mathcal{A}_e \neq \emptyset\}| + k(|V| - |\mathcal{P}|) : \mathcal{P} \text{ a partition of } V\}. \quad (2.4.5)$$

Proof. Let \mathcal{Z}' be obtained from \mathcal{Z} by deleting all but one element of $\mathcal{Z} \cap \mathcal{A}_e$ for all $e \in \mathcal{E}$ with $|\mathcal{Z} \cap \mathcal{A}_e| \geq 2$. As all elements in \mathcal{A}_e are parallel in $M_{\mathcal{F}}^k$, we obtain that $r_{M_{\mathcal{F}}^k}(\mathcal{Z}) = r_{M_{\mathcal{F}}^k}(\mathcal{Z}')$. As $|\mathcal{Z}' \cap \mathcal{A}_e| \leq 1$ for every $e \in \mathcal{E}$, there exists a matroid M' that is isomorphic to $M_{\mathcal{H}}^k$, is a restriction of $M_{\mathcal{F}}^k$ and whose ground set contains \mathcal{Z}' . It follows from Theorem 2.4.8 that

$$\begin{aligned} r_{M_{\mathcal{F}}^k}(\mathcal{Z}) &= r_{M_{\mathcal{F}}^k}(\mathcal{Z}') \\ &= r_{M'}(\mathcal{Z}') \\ &= \min\{|\mathcal{Z}' \cap \mathcal{A}(\mathcal{P})| + |\mathcal{Z}' \cap \mathcal{E}(\mathcal{P})| + k(|V| - |\mathcal{P}|) : \mathcal{P} \text{ a partition of } V\} \\ &= \min\{|\mathcal{Z} \cap \mathcal{A}(\mathcal{P})| + |\{e \in \mathcal{E}(\mathcal{P}) : \mathcal{Z} \cap \mathcal{A}_e \neq \emptyset\}| + k(|V| - |\mathcal{P}|) : \mathcal{P} \text{ a partition of } V\}. \end{aligned}$$

\square

Again, for the algorithmic part, we need to show that an independence oracle for $M_{\mathcal{F}}^k$ is available.

Lemma 2.4.4. *Given a mixed hypergraph $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ and $\mathcal{Z} \subseteq \mathcal{A} \cup \mathcal{A}_{\mathcal{E}}$, we can decide in polynomial time if \mathcal{Z} is independent in $M_{\mathcal{F}}^k$.*

Proof. If \mathcal{Z} contains at least 2 elements of \mathcal{A}_e for some $e \in \mathcal{E}$, then \mathcal{Z} is dependent in $M_{\mathcal{F}}^k$ by definition. Otherwise, there is a matroid M' that is isomorphic to $M_{\mathcal{H}_{\mathcal{F}}}^k$, is a restriction of $M_{\mathcal{F}}^k$ and whose ground set contains \mathcal{Z} . Further, M' can be found efficiently. It therefore suffices to prove that a polynomial time independence oracle for $M_{\mathcal{H}_{\mathcal{F}}}^k$ is available. This follows immediately from Lemma 2.4.3 and Theorem 2.4.5. \square

2.4.2.3 Tanigawa matroids

In order to deal with this class of matroids, we first introduce some more notation that will prove useful in Section 4. A *(simply) matroid-rooted digraph* is a tuple (D, M) where $D = (V \cup R, A)$ is a (simply) rooted digraph and $M = (R, r_M)$ is a matroid with ground set R and rank function r_M .

A *(simply) matroid-rooted mixed hypergraph* is a tuple (\mathcal{F}, M) where $\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E})$ is a (simply) rooted mixed hypergraph and $M = (R, r_M)$ is a matroid with ground set R and rank function r_M . Note that a rooted mixed hypergraph can be considered as a matroid-rooted mixed hypergraph for the free matroid on R .

2.4.2.3.1 Tanigawa matroids from simply matroid-rooted hypergraphs

This matroid is a straightforward extension of the construction of the graphic case found by Király, Szigeti and Tanigawa in [79]. The authors of [79] also rely on ideas of Katoh and Tanigawa, see [74]. Given a grounded simply rooted mixed hypergraph $(\mathcal{H} = (V \cup R, \delta_{\mathcal{H}}^+(R) \cup \mathcal{E}), M)$, we define $\mathcal{I}_{(\mathcal{H}, M)} = \{\mathcal{X} \subseteq \delta_{\mathcal{H}}^+(R) \cup \mathcal{E} : r_M(R)(|V(\mathcal{Y})| - 1) + r_M(R(\mathcal{Y})) \geq |\mathcal{Y}| \text{ for all nonempty } \mathcal{Y} \subseteq \mathcal{X}\}$. The following is proven for the graphic case in a slightly different form in [79]. Its proof can be literally generalized for the hypergraphic case.

Theorem 2.4.9. $\mathcal{I}_{(\mathcal{H}, M)}$ is the set of independent sets of a matroid on $\delta_{\mathcal{H}}^+(R) \cup \mathcal{E}$.

We refer to this matroid as the *Tanigawa matroid* $\mathbf{T}_{(\mathcal{H}, M)}$ of (\mathcal{H}, M) .

As mentioned in [79], the following has been shown in [74].

Lemma 2.4.5. A polynomial time independence oracle for $T_{(\mathcal{H}, M)}$ is available.

2.4.2.3.2 Extended Tanigawa matroids from simply matroid-rooted mixed hypergraphs

We now extend the previous construction to matroid-rooted mixed hypergraphs. Let $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}), M)$ be a matroid-rooted mixed hypergraph, let $\mathcal{D}_{\mathcal{F}} = (V \cup R, \mathcal{A} \cup \mathcal{A}_{\mathcal{E}})$ be the directed extension of \mathcal{F} and let $\mathcal{H} = (V \cup R, \delta_{\mathcal{F}}^+(R) \cup \mathcal{E}_{\mathcal{A} - \delta_{\mathcal{F}}^+(R)} \cup \mathcal{E})$ be the underlying grounded rooted mixed hypergraph of \mathcal{F} . The ground set of the *extended Tanigawa matroid* of (\mathcal{F}, M) is $\mathcal{A} \cup \mathcal{A}_{\mathcal{E}}$ and it is constructed from $T_{(\mathcal{H}, M)}$ by replacing every $e \in \mathcal{E}$ by $|e|$ parallel copies of itself, associating them to the elements in \mathcal{A}_e and associating every element of $\mathcal{E}_{\mathcal{A} - \delta_{\mathcal{F}}^+(R)}$ to the corresponding element of $\mathcal{A} - \delta_{\mathcal{F}}^+(R)$. We will refer to this matroid as $\mathbf{T}_{(\mathcal{F}, M)}$. The following result can be concluded from Lemma 2.4.5 by similar means as those used when proving Lemma 2.4.4.

Lemma 2.4.6. A polynomial time independence oracle for $T_{(\mathcal{F}, M)}$ is available.

Chapter 3

Orientations

In this section, we deal with problems on the relationship between the connectivity properties of a given graph and the connectivity properties of its orientations. More particularly, given a certain connectivity property in directed graphs, we wish to determine which graphs have an orientation with this property.

This section is divided into two parts. The first one is concerned with orientations for arc-connectivity and the second one is concerned with orientations for vertex-connectivity.

The tractability of these two problem settings is very different. In the case of arc-connectivity, for any $k \in \mathbb{Z}_{>0}$, a characterization of graphs admitting a k -arc-connected orientation is well-known due to a result of Nash-Williams [96]. We hence discuss several related more complicated problems in Section 3.1. Firstly, we deal with a result of Nash-Williams that is much stronger than the above characterization and give some results related to it. Next, we deal with a more general setting concerning orientations for local arc-connectivity. After, we consider a problem where we search for an orientation for which in addition to the connectivity condition, conditions on the parity of the in-degree of every vertex are imposed. Finally, we deal with orientations of a graph that form an intermediate step between strong connectivity and 2-arc-connectivity.

The situation for vertex-connectivity is much different. A characterization of highly vertex-connected graphs seems out of reach due to a hardness result of Durand de Gevigney [24]. We give a survey on related results. Further, we determine some small classes of graphs all of whose Eulerian orientations are highly vertex-connected.

3.1 Orientations for arc-connectivity

This section is dedicated to finding an orientation of a given graph that satisfies some arc-connectivity requirements. For rooted connectivity, it is easy to see that every rooted connected graph $G = (V \cup r, E)$ has a root-connected orientation. The following generalization to arbitrary k was proven by Frank in [44].

Theorem 3.1.1. *Given some positive integer k and a singularly rooted graph $G = (V \cup r, E)$, there is a k -root-connected orientation of G if and only if $e_G(\mathcal{P}) \geq k(|\mathcal{P}| - 1)$ for every partition \mathcal{P} of $V \cup r$.*

We now turn our attention to global arc-connectivity. The following fundamental theorem characterizes graphs having a strongly connected orientation. It was proven by Robbins [100] and dates back to 1939.

Theorem 3.1.2. *A graph has a strongly connected orientation if and only if it is 2-edge-connected.*

In 1960, this was generalized to higher connectivity by Nash-Williams [96].

Theorem 3.1.3. *For every positive integer k , a graph has a k -arc-connected orientation if and only if it is $2k$ -edge-connected.*

Theorems 3.1.2 and 3.1.3 can be considered the starting point of the theory of graph orientations for connectivity. In the last decades, numerous generalizations of Theorems 3.1.2 and 3.1.3 have been found.

One such attempt is the consideration of more general objects like mixed graphs and hypergraphs instead of graphs. Concerning hypergraphs, Theorem 3.1.3 can be generalized in the following way:

Theorem 3.1.4. *A hypergraph $\mathcal{H} = (V, \mathcal{E})$ has a k -dyperedge-connected orientation if and only if $e_{\mathcal{E}}(\mathcal{P}) \geq k|\mathcal{P}|$ holds for every partition \mathcal{P} of V .*

While for $k = 1$ a relatively simple algorithmic proof can be found in [41], a more sophisticated technique is applied to obtain the more general result by Frank, Király and Király in [50].

For mixed graphs, the following generalization of Theorem 3.1.2 was observed by Boesch and Tindal [17]. A rather simple algorithmic proof was given by Frank [43].

Theorem 3.1.5. *A mixed graph $G = (V, A \cup E)$ has a strongly connected orientation if and only if the underlying graph of G is 2-edge-connected and there is no nonempty, proper $X \subseteq V$ with $d_E(X) = d_A^-(X) = 0$.*

It turns out that the problem becomes much more complicated when considering k -arc-connected orientations for arbitrary positive integers k . Nevertheless, a pretty technical characterization can be obtained as pointed out in Section 16.1.5 of [41]. A generalization of this result to mixed hypergraphs by Frank, Király and Király can be found in [50]

In this section, we discuss several further extensions of Theorems 3.1.2 and Theorem 3.1.3.

First, in Section 3.1.1, we discuss a much stronger form of Theorem 3.1.3 which was also proven by Nash-Williams in [96]. We survey some results related to this theorem and its proof and add a new negative result.

Section 3.1.2 is concerned with the very general problem of finding orientations with a prescribed local arc-connectivity. We show that this problem is hard in general and give some related open problems.

In Section 3.1.3, we consider a different way of generalizing Theorems 3.1.2 and 3.1.3. We search for an orientation in which, in addition to the arc-connectivity property, the indegree of every vertex is required to be of a certain parity. The most interesting question in this field is whether the existence of a strongly connected orientation with this additional property can be decided in polynomial time. While this question remains open, we reduce it to a special case where all vertices are of small degree. Further, we give a negative result for the corresponding problem in mixed graphs.

In Section 3.1.4, we deal with a different problem motivated by Theorems 3.1.2 and Theorem 3.1.3. Theorem 3.1.3 raises the question whether graphs of odd global edge-connectivity also have some interesting orientability properties. We approach this by allowing several orientations for a given graph and requiring that every edge is insignificant in a certain way in some of these orientations. This leads to the definition of a graph invariant that counts the number of necessary orientations. Concentrating on a base case of this problem, we provide several bounds on this parameter.

3.1.1 Well-balanced orientations

This section deals with a much stronger version of Theorem 3.1.3 that was also proven by Nash-Williams [96]. In Section 3.1.1.1, we describe this theorem and give a rough overview of its proof. In Section 3.1.1.2 we give an overview of possible extensions of the theorem. Finally, in Section 3.1.1.3, we give a hardness result which is related to the proof of the theorem.

3.1.1.1 Nash-Williams' theorem

While Theorem 3.1.3 gives a necessary and sufficient condition for a graph admitting an orientation of high global arc-connectivity, it does not take into account the fact that the connectivity properties of a digraph can only be described very roughly by a single value displaying its global arc-connectivity. For example, if the connectivity between two vertices in a given graph is high, the connectivity in the orientation between these vertices which is guaranteed by Theorem 3.1.3 can be small due to a small cut in the given graph even if this cut does not separate the vertices. In order to overcome this inconvenience, a much stronger connectivity condition for the orientation has been introduced by Nash-Williams. Given a graph $G = (V, E)$, an orientation \vec{G} of G is called *well-balanced* if $\lambda_{\vec{G}}(u, v) \geq \lfloor \frac{\lambda_G(u, v)}{2} \rfloor$ for all $u, v \in V$. The famous theorem of Nash-Williams states that such an orientation can always be found. It dates back to 1960 and can be found in [96].

Theorem 3.1.6. *Every graph has a well-balanced orientation.*

Despite numerous attempts, no simple proof of Theorem 3.1.6 has been found. All proofs rely on an intermediate result which states that it is possible to add a set of edges to a given graph such that the graph becomes Eulerian without changing the connectivity properties of the graph too much. More concretely, given a graph $G = (V, E)$, an *odd-vertex pairing* of G is another graph $F = (V, E')$ on the same vertex set such that every vertex that is of odd degree in G is of degree 1 in F and every vertex that is of even degree in G is an isolated vertex of F . The point of using an odd-vertex pairing F is that we hope to find an Eulerian orientation of $G \cup F$ whose restriction to G is a well-balanced orientation of G . We say that an odd-vertex pairing is *orientation-admissible* if for every Eulerian orientation $\vec{G} \cup \vec{F}$ of $G \cup F$, we have that \vec{G} is a well-balanced orientation of G . Observe that the existence of an orientation-admissible odd-vertex pairing clearly implies a well-balanced orientation. The approach of Nash-Williams relies on proving the existence of an odd-vertex pairing with a slightly stronger property. Given a graph $G = (V, E)$, an odd-vertex pairing F is called *cut-admissible* if $d_F(X) \leq d_G(X) - R_G(X)$ for all $X \subseteq V$ where $R_G(X) = \max\{\lfloor \frac{\lambda_G(u, v)}{2} \rfloor : u \in X, v \in V - X\}$. This definition is motivated by the following result which is part of the proof of Nash-Williams in [96]. As the proof is very simple, we include it here.

Lemma 3.1.1. *Let $G = (V, E)$ be a graph and let F be an odd-vertex pairing of G . If F is cut-admissible, then F is also orientation-admissible.*

Proof. We show that if F is not orientation-admissible, then F is also not cut-admissible. Suppose that F is not orientation-admissible, so there is an Eulerian orientation $\vec{G} \cup \vec{F}$ of $G \cup F$ such that \vec{G} is not well-balanced. This means that there are some $u, v \in V$ such that $\lambda_{\vec{G}}(u, v) < \lfloor \frac{\lambda_G(u, v)}{2} \rfloor$. Therefore there is some $u\bar{v}$ -set $X \subsetneq V$ such that $d_{\vec{G}}^+(X) < \lfloor \frac{\lambda_G(u, v)}{2} \rfloor$. As $\vec{G} \cup \vec{F}$ is Eulerian, we obtain that $d_F(X) \geq d_{\vec{G}}^-(X) - d_{\vec{G}}^+(X) = d_G(X) - 2d_{\vec{G}}^+(X) > d_G(X) - 2\lfloor \frac{\lambda_G(u, v)}{2} \rfloor \geq d_G(X) - R_G(X)$, so F is not cut-admissible. \square

On the other hand, not every orientation-admissible odd-vertex pairing is cut-admissible, see Figure 3.1.1. The key ingredient of the proof of Theorem 3.1.6 is the following result

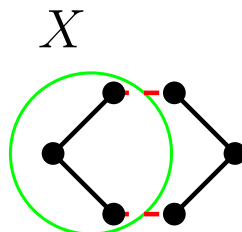


Figure 3.1.1: The edges of G are marked in solid and those of F are marked in dashed. The set X shows that F is not cut-admissible. On the other hand, F is trivially orientation-admissible.

that asserts the existence of a cut-admissible pairing for every graph.

Theorem 3.1.7. *Every graph has a cut-admissible odd-vertex pairing.*

Observe that by Lemma 3.1.1, Theorem 3.1.7 implies Theorem 3.1.6. Taken into account the simplicity of the proof of Lemma 3.1.1, the main difficulty of the proof of Theorem 3.1.6 clearly is the proof of Theorem 3.1.7. The original proof of Nash-Williams is based on an inductive argument and is very complicated. New proofs have been provided by Frank [39] and Mader [89], but all these proofs remain pretty involved.

Further, Nash-Williams observed that his proof technique allows to obtain orientations that have an even stronger property than being well-balanced. Given a graph $G = (V, E)$, an orientation \vec{G} of G is called *best-balanced* if \vec{G} is well-balanced and $|d_{\vec{G}}^+(v) - d_{\vec{G}}^-(v)| \leq 1$ holds for all $v \in V$. Observe that not every well-balanced orientation is best-balanced, see Figure 3.1.2.



Figure 3.1.2: It is easy to see that the digraph is a well-balanced orientation of its underlying graph. However, $|d_{\vec{G}}^+(v) - d_{\vec{G}}^-(v)| = 2$, so the orientation is not best-balanced.

Clearly, the proof technique described above using orientation-admissible pairings yields a well-balanced orientation that is in fact best-balanced. This observation, together with Theorem 3.1.7 yields the following result:

Theorem 3.1.8. [96] *Every graph has a best-balanced orientation.*

Next, we wish to mention that well-balanced and best-balanced orientations can also be found efficiently. In particular, Gabow [52] provided an algorithm that finds a cut-admissible odd-vertex pairing in a given graph and runs in $O(nm^2)$ time. Together with the proof technique described above, this yields an algorithm of the same running time that finds a best-balanced orientation.

3.1.1.2 Extensions of well-balanced pairings

This part gives an overview of possible extensions of Theorems 3.1.6 and 3.1.8. While it contains a lot of negative results, it also contains several interesting extensions of Theorems 3.1.6 and 3.1.8 and some open questions.

As a first extension, weighted versions can be considered. Most generally, for every edge, we can associate a cost to each of its orientations. For some graph $G = (V, E)$, recall that \mathbf{A}_E is the set of arcs described by $A_E = \{uv, vu : uv \in E\}$. Observe that

for every orientation $\vec{G} = (V, \vec{E})$ of G , we have $\vec{E} \subseteq A_E$. We consider the following two problems:

Min Cost Well-Balanced orientation (MCWBO):

Instance: A graph $G = (V, E)$, a cost function $c : A_E \rightarrow \mathbb{R}$ and a constant $k \in \mathbb{R}$.

Question: Is there a well-balanced orientation $\vec{G} = (V, \vec{E})$ of G such that $c(\vec{E}) < k$?

Min Cost Best-Balanced orientation (MCBBO):

Instance: A graph $G = (V, E)$, a cost function $c : A_E \rightarrow \mathbb{R}$ and a constant $k \in \mathbb{R}$.

Question: Is there a best-balanced orientation $\vec{G} = (V, \vec{E})$ of G such that $c(\vec{E}) < k$?

The following negative result is proven by Bernáth et al. in [13].

Theorem 3.1.9. *MCWBO and MCBBO are NP-complete.*

In a more restricted setting, costs can be associated to the vertices rather than to the possible orientations. We consider the following two problems:

Min Vertex Cost Well-Balanced Orientation (MVCWBO):

Instance: A graph $G = (V, E)$, a cost function $c : V \rightarrow \mathbb{R}$ and a constant $k \in \mathbb{R}$.

Question: Is there a well-balanced orientation \vec{G} of G such that $\sum_{v \in V} c(v) d_{\vec{G}}^-(v) < k$?

Min Vertex Cost Best-Balanced Orientation (MVCBBO):

Instance: A graph $G = (V, E)$, a cost function $c : V \rightarrow \mathbb{R}$ and a constant $k \in \mathbb{R}$.

Question: Is there a best-balanced orientation \vec{G} of G such that $\sum_{v \in V} c(v) d_{\vec{G}}^-(v) < k$?

Observe that MVCWBO is a special case of MCWBO as the cost of each arc in A_E can be chosen to be the cost associated to its head. Similarly, MVCBBO is a special case of MCBBO. Nevertheless, it is proven in [13] that even these more restricted problems are not algorithmically tractable.

Theorem 3.1.10. *MVCWBO and MVCBBO are NP-complete.*

A next attempt to generalize Theorems 3.1.6 and 3.1.8 is to search for well-balanced or best-balanced orientations that satisfy some restrictions on the in- and outdegree of each vertex. We consider the following two problems:

Bounded Well-Balanced Orientation (BWBO):

Instance: A graph $G = (V, E)$ and two functions $l^+, l^- : V \rightarrow \mathbb{Z}_{\geq 0}$.

Question: Is there a well-balanced orientation $\vec{G} = (V, \vec{E})$ of G such that $d_{\vec{G}}^+(v) \geq l^+(v)$ and $d_{\vec{G}}^-(v) \geq l^-(v)$?

Bounded Best-Balanced Orientation (BBBO):

Instance: A graph $G = (V, E)$ and two functions $l^+, l^- : V \rightarrow \mathbb{Z}_{\geq 0}$.

Question: Is there a best-balanced orientation $\vec{G} = (V, \vec{E})$ of G such that $d_{\vec{G}}^+(v) \geq l^+(v)$ and $d_{\vec{G}}^-(v) \geq l^-(v)$?

Again, the algorithmic tractability of these problems has been disproven in [13].

Theorem 3.1.11. *BWBO and BBBO are NP-complete.*

We will make use of the NP-completeness of BWBO in Section 3.1.2 for a reduction.

A similar problem that is still open can be obtained when only one bound is imposed on the degrees of the orientation. In particular, the following question was asked by Frank in [41] as Research Problem 9.8.2, see also [33].

Research Problem 3.1.1. Let $G = (V, E)$ be a graph and $l^- : V \rightarrow \mathbb{Z}_{\geq 0}$ be a function. Can we check in polynomial if there is a best-balanced orientation \vec{G} of G such that $d_{\vec{G}}^-(v) \geq l^-(v)$ for all $v \in V$?

Next, we try to find well-balanced or best-balanced orientations of a given graph G that maintain these properties when considering their restrictions to a subgraph of G . If G is a graph, H is a subgraph of G and \vec{G} is an orientation of G , we denote by $\vec{H}_{\vec{G}}$ the orientation of H that is obtained by orienting all edges of H in the same way they are oriented in \vec{G} . Taking into account the number of related hard problems, it is perhaps surprising that Nash-Williams was able to establish the following result along with Theorem 3.1.6 in [96].

Theorem 3.1.12. Let G be a graph and let H be a subgraph of G . Then G has a best-balanced orientation \vec{G} such that $\vec{H}_{\vec{G}}$ is a best-balanced orientation of H .

However, we cannot hope to get a similar result for a longer chain of nested subgraphs as was proven in [13].

Theorem 3.1.13. There are 3 graphs G_1, G_2, G_3 such that G_2 is a subgraph of G_1 , G_3 is a subgraph of G_2 and there is no well-balanced orientation \vec{G}_1 of G_1 such that $\vec{G}_2_{\vec{G}_1}$ and $\vec{G}_3_{\vec{G}_1}$ are well-balanced.

On the other hand, in [81], Király and Szigeti establish several strong results on simultaneous well-balanced orientations. First, they generalize Theorem 3.1.12 for the case when we want the orientation to be well-balanced on an arbitrary number of edge-disjoint subgraphs.

Theorem 3.1.14. Let $G = (V, E)$ and let $G_1 = (V, E_1), \dots, G_t = (V, E_t)$ be a collection of subgraphs of G such that $E_i \cap E_j = \emptyset$ for all $1 \leq i < j \leq t$. Then there is a best-balanced orientation \vec{G} of G such that $\vec{G}_{i_{\vec{G}}}$ is best-balanced for all $i = 1, \dots, t$.

They also establish a result that shows that for Eulerian graphs orientations are available which maintain the property of being best-balanced when deleting an arbitrary vertex.

Theorem 3.1.15. [81] Let $G = (V, E)$ be an Eulerian graph. Then there is an Eulerian orientation \vec{G} of G such that $\vec{G} - v$ is a best-balanced orientation of $G - v$ for all $v \in V$.

They also give another result that shows that we can find best-balanced orientations that maintain that property when certain vertex sets are contracted.

Theorem 3.1.16. [81] Let $G = (V, E)$ be a graph and let $\{V_1, \dots, V_t\}$ be a partition of V . Then G has a best-balanced orientation \vec{G} such that $\vec{G}/(V - V_i)$ is a best-balanced orientation of $G/(V - V_i)$ for $i = 1, \dots, t$ and $((\vec{G}/V_1)/\dots)/V_t$ is a best-balanced orientation of $((G/V_1)/\dots)/V_t$.

Our next attempt of generalization is based on the consideration of mixed graphs instead of undirected graphs. First we consider the question whether a given mixed graph can be oriented to obtain a well-balanced orientation of the underlying graph. More concretely, we consider the following two problems:

Mixed well-balanced orientation (MWBO):

Instance: A mixed graph $G = (V, A \cup E)$.

Question: Is there an orientation \vec{G} of G that is a well-balanced orientation of the underlying graph of G ?

Mixed best-balanced orientation (MBBO):

Instance: A mixed graph $G = (V, A \cup E)$.

Question: Is there an orientation \vec{G} of G that is a best-balanced orientation of the underlying graph of G ?

Unfortunately, it turns out again that these problems are not algorithmically tractable. The following result is proven in [13].

Theorem 3.1.17. *MBBO is NP-complete.*

The following result is due to Bernáth and Joret [14].

Theorem 3.1.18. *MWBO is NP-complete.*

We next propose an open problem making a different connection between well-balanced orientations and mixed graphs. Given a mixed graph $G = (V, A \cup E)$, we define $\lambda_G(\mathbf{u}, \mathbf{v})$ to be the maximum number of edge- and arc-disjoint mixed paths from u to v and let $\lambda_G^*(\mathbf{u}, \mathbf{v}) = \min\{\lambda_G(u, v), \lambda_G(v, u)\}$. We say that an orientation \vec{G} of G is *well-balanced* if $\lambda_{\vec{G}}(u, v) \geq \lfloor \frac{\lambda_G^*(u, v)}{2} \rfloor$ for all $u, v \in V$. We here propose the following conjecture.

Conjecture 3.1.1. *Every mixed graph has a well-balanced orientation.*

Observe that for undirected graphs, by Theorem 2.1.2, Conjecture 3.1.1 is equivalent to Theorem 3.1.6 and for directed graphs, Conjecture 3.1.1 trivially holds. Unfortunately, no good equivalent of best-balanced orientations seems to exist for mixed graphs.

Finally, one can also consider well-balanced orientations of hypergraphs. Clearly, not every hypergraph has a well-balanced orientation as can be seen by considering the hypergraph that contains 3 vertices and whose hyperedge set consists of two copies of the vertex set. It therefore remains to determine which hypergraphs have a well-balanced orientation.

Research Problem 3.1.2. *Determine the complexity of deciding whether a hypergraph has a well-balanced orientation.*

In [35], it is mentioned that further interesting questions could be obtained by changing the notion of dyperedges.

3.1.1.3 Checking the admissibility of odd-vertex pairings

This section provides a negative result concerning the algorithmic verification of the admissibility properties of an odd-vertex pairing.

Frank asked whether the admissibility properties of a given odd-vertex pairing can be checked in polynomial time. This problem can be found as Research Problem 9.8.1 in [41]. We here give a negative answer to this question following the proof in [59]. More formally, we consider the following two problems:

Cut-admissibility (CA):

Instance: A graph G and an odd-vertex pairing F of G .

Question: Is F cut-admissible in G ?

Orientation-admissibility (OA):

Instance: A graph G and an odd-vertex pairing F of G .

Question: Is F orientation-admissible in G ?

While it is not clear whether CA and OA are in NP, they can easily be seen to be in co-NP. As our main results of this section, we prove the following two theorems.

Theorem 3.1.19. *CA is co-NP-complete.*

Theorem 3.1.20. *OA is co-NP-complete.*

In Subsection 3.1.1.3.1, we introduce a gadget we will need later in the main proof and give some of its properties. In Subsection 3.1.1.3.2, we show through a reduction from AMAXCUT that an intermediate problem with a more local cut condition than CA is hard. In Subsection 3.1.1.3.3, we give the construction of the graph that is the key in our reduction. In Subsection 3.1.1.3.4, we conclude that the reduction works indeed.

3.1.1.3.1 Augmented (α, β) -grids

In this section, we introduce a class of grid-like graphs which will be used as a gadget in our reduction. A *grid* is a graph on ground set $\{1, \dots, \mu\} \times \{1, \dots, \nu\}$ for some positive integers μ, ν where two vertices (i_1, j_1) and (i_2, j_2) are adjacent if $|i_1 - i_2| + |j_1 - j_2| = 1$. For some $i \in \{1, \dots, \mu\}$, we call $\{(i, 1), \dots, (i, \nu)\}$ the *row* i . Similarly, for some $j \in \{1, \dots, \nu\}$, we call $\{(1, j), \dots, (\mu, j)\}$ the *column* j .

In order to define augmented (α, β) -grids for an odd integer $\alpha \geq 3$ and an integer $\beta \geq 2$, we first consider a grid with $\alpha\beta$ rows and $\frac{\alpha+1}{2}$ columns. Now, for some $1 \leq \gamma \leq \beta$, let $\mathbf{L}_\gamma = \{l_1, \dots, l_\gamma\} = \{(\alpha, 1), (2\alpha, 1), \dots, (\gamma\alpha, 1)\}$ and $\mathbf{P}_\gamma = \{p_1, \dots, p_\gamma\} = \{(\alpha, \frac{\alpha+1}{2}), (2\alpha, \frac{\alpha+1}{2}), \dots, (\gamma\alpha, \frac{\alpha+1}{2})\}$. We use \mathbf{L} for L_β and \mathbf{P} for P_β . We now create the *augmented (α, β) -grid* \mathbf{W} by adding an edge from $(1, j)$ to $(\alpha\beta, j)$ for all $j = 1, \dots, \frac{\alpha+1}{2}$ and by adding parallel edges in the columns 1 and $\frac{\alpha+1}{2}$ in a way that none of them is incident to a vertex in $L \cup P$ and that every vertex in $V(\mathbf{W}) - (L \cup P)$ has degree 4 in \mathbf{W} . Observe that this is possible because both $\alpha - 1$ and $\alpha + 1$ are even. An example can be found in Figure 3.1.3.

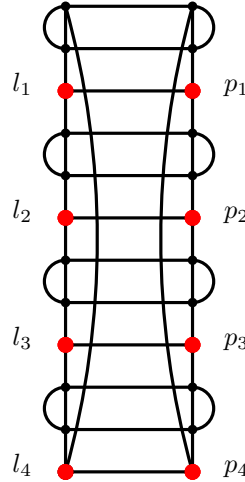


Figure 3.1.3: An augmented $(3, 4)$ -grid.

Later, when W is not clear from the context, we use $\mathbf{L}(\mathbf{W})$ for the set L etc. We now collect some properties of augmented (α, β) -grids.

Lemma 3.1.2. *Let $W = (V, E)$ be an augmented (α, β) -grid for some odd integer $\alpha \geq 3$ and some integer $\beta \geq 2$. Then W is 3-edge-connected and if $d_W(X) = 3$ for some nonempty $X \subsetneq V$, then $X = \{v\}$ or $X = V - \{v\}$ for some $v \in L(W) \cup P(W)$.*

Proof. Let $\emptyset \subsetneq X \subsetneq V$ such that $d_W(X) \leq 3$. Observe that every row that intersects both X and $V - X$ contributes at least 1 to $d_W(X)$ and every column that intersects both X and $V - X$ contributes at least 2 to $d_W(X)$. It follows that one of X or $V - X$ is contained in one row and one column. We obtain that $|X| = 1$ or $|V - X| = 1$ and so the statement follows by construction. \square

Lemma 3.1.3. *Let $W = (V, E)$ be an augmented (α, β) -grid for some odd integer $\alpha \geq 3$ and some integer $\beta \geq 2$. Further, let $X \subseteq V$ such that both $W[X]$ and $W[V - X]$ have a connected component containing at least two vertices of $L(W) \cup P(W)$. Then $d_W(X) > \alpha$.*

Proof. Suppose for the sake of a contradiction that there is some $X \subseteq V$ such that both $W[X]$ and $W[V - X]$ have a connected component containing at least two vertices of $L(W) \cup P(W)$ and $d_W(X) \leq \alpha$. We choose X so that the total number of connected components of $W[X]$ and $W[V - X]$ is minimized. First suppose that $W[X]$ is disconnected. It follows from the assumption that $W[X]$ has a connected component with vertex set C such that $W[X - C]$ has a connected component containing at least two vertices in $L(W) \cup P(W)$. Let $X' = X - C$. We obtain $d_W(X') \leq d_W(X) \leq \alpha$, a contradiction to the minimal choice of X . It follows that $W[X]$ is connected. Similarly, $W[V - X]$ is connected.

If every column contains an element of X and an element of $V - X$, each column contributes 2 to $d_W(X)$ and so $d_W(X) \geq 2 \frac{\alpha+1}{2} > \alpha$. We may hence suppose by symmetry that there is a column that is completely contained in X and that there are two vertices $l_{i_1}, l_{i_2} \in (V - X) \cap L$. Observe that every path from l_{i_1} to l_{i_2} intersects at least $\min\{|i_1 - i_2|, \beta - (|i_1 - i_2|)\} \alpha + 1 > \alpha$ rows. Each of these rows contributes 1 to $d_W(X)$, so $d_W(X) > \alpha$. \square

3.1.1.3.2 The intermediate cut problem

Let $(H = (V_H, E_H), k)$ be an instance of AMAXCUT. We abbreviate $|V_H|$ and $|E_H|$ to n and m , respectively. Let $M = mn - k$. We now create a graph $G_1 = (V_1, E_1)$ with $V_1 = V_H \cup \{q, s, t\}$ where q, s and t are 3 new vertices. Let E_1 consist of M edges from q to s , m edges from s to every $v \in V_H$ and m edges from t to every $v \in V_H$. A schematic drawing of G_1 can be found in Figure 3.1.4.

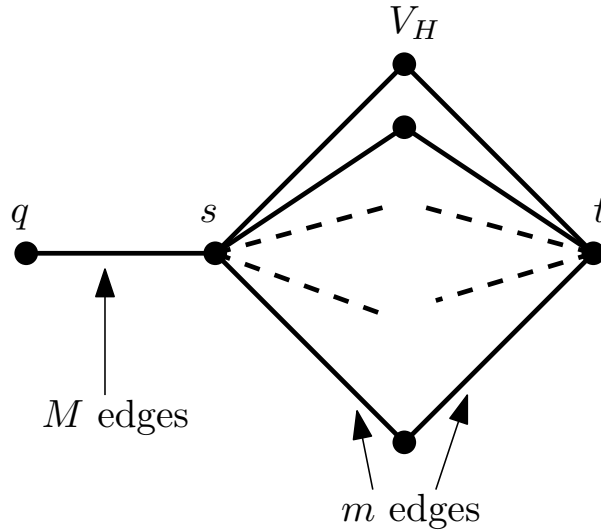


Figure 3.1.4: A schematic drawing of G_1 .

Lemma 3.1.4. *There is some $q\bar{t}$ -set $X \subseteq V_1$ such that $d_{G_1}(X) - d_H(X \cap V_H) < M$ if and only if (H, k) is a positive instance of AMAXCUT.*

Proof. First suppose that (H, k) is a positive instance of AMAXCUT, so there is some $X \subseteq V_H$ such that $d_H(X) > k$. Let $X' = \{q, s\} \cup X$. Observe that X' is a $q\bar{t}$ -set and $d_{G_1}(X') = mn$. This yields $d_{G_1}(X') - d_H(X' \cap V_H) = d_{G_1}(X') - d_H(X) < M$.

Now suppose that there is some $q\bar{t}$ -set $X \subseteq V_1$ such that $d_{G_1}(X) - d_H(X \cap V_H) < M$.

Claim 3.1.1. $s \in X$.

Proof. Suppose otherwise. If $X = \{q\}$, then $d_{G_1}(X) - d_H(X \cap V_H) = M - 0 \not\leq M$, a contradiction. We may hence suppose that X contains some $v \in V_H$. It follows from $d_H(X \cap V_H) \leq m$ and construction that $d_{G_1}(X) - d_H(X \cap V_H) \geq d_{G_1}(q, s) + d_{G_1}(v, t) - m = M + m - m \not\leq M$, a contradiction. \square

By Claim 3.1.1 and construction, we obtain $d_{G_1}(X) = mn$. This yields $d_H(X \cap V_H) > d_{G_1}(X) - M = mn - M = k$, so (H, k) is a positive instance of AMAXCUT. \square

3.1.1.3.3 The main construction

We now construct an instance (G_2, F) of CA. The graph $\mathbf{G}_2 = (\mathbf{V}_2, \mathbf{E}_2)$ is obtained from G_1 by replacing all vertices in $V_1 - \{q, t\}$ by certain gadgets.

For every $v \in V_H$, G_2 contains an augmented $(M + m + 1, m + \frac{d_H(v)}{2})$ -grid W^v . Further, G_2 contains an augmented $(M + m + 1, M + \frac{k}{2})$ -grid W^s . Observe that W^v for all $v \in V_H$ and W^s are well-defined because m, k, M and $d_H(v)$ for all $v \in V_H$ are even. Let $V_2 = \cup_{v \in V_H} V(W^v) \cup V(W^s) \cup \{q, t\}$. We now add an edge from q to each vertex in $L_M(W^s)$. We next add a perfect matching between $(L(W^s) - L_M(W^s)) \cup P(W^s)$ and $\cup_{v \in V_H} L_m(W^v)$. Observe that this is possible because $|(L(W^s) - L_M(W^s)) \cup P(W^s)| = \frac{k}{2} + M + \frac{k}{2} = mn = |\cup_{v \in V_H} L_m(W^v)|$. Finally, we add an edge from every vertex in $\cup_{v \in V_H} P_m(W^v)$ to t . Observe that G_1 can be obtained from G_2 by contracting each W^v and W^s into single vertices.

We now prove an important property of G_2 .

Lemma 3.1.5. *For any $\emptyset \subsetneq X \subsetneq V_2$, we have*

$$R_{G_2}(X) = 2 \lfloor \frac{\min\{\max\{d_{G_2}(v) : v \in X\}, \max\{d_{G_2}(v) : v \in V_2 - X\}\}}{2} \rfloor.$$

Proof. As G_1 is 4-edge-connected and by Lemma 3.1.2 applied to W^s and W^v for all $v \in V_H$, we obtain that $\lambda_{G_2}(u, v) = \min\{d_{G_2}(u), d_{G_2}(v)\}$ for all $u, v \in V_2$ with $\{u, v\} \neq \{q, t\}$. This shows the statement for all $\emptyset \subsetneq X \subsetneq V_2$ such that $\{q, t\} \subseteq X$ or $\{q, t\} \subseteq V_2 - X$. On the other hand, if X is a $q\bar{t}$ -set or a $t\bar{q}$ -set, we have $\min\{\max\{d_{G_2}(v) : v \in X\}, \max\{d_{G_2}(v) : v \in V_2 - X\}\} = M$. As M is even, it hence suffices to prove that $\lambda_{G_2}(q, t) = M$.

We have $\lambda_{G_2}(q, t) \leq d_{G_2}(q) = M$. Next, there is an edge between q and $l_{j_1}(W^s)$ for all $j_1 = 1, \dots, M$ which can be concatenated to a path from $l_{j_1}(W^s)$ to $p_{j_1}(W^s)$ using only vertices of a single row of W^s . Now there is an edge from $p_{j_1}(W^s)$ to a vertex $l_{j_2}(W^v)$ for some $j_2 \in \{1, \dots, m\}$ and some $v \in V_H$. Finally, there is a path from $l_{j_2}(W^v)$ to $p_{j_2}(W^v)$ and an edge from $p_{j_2}(W^v)$ to t . This yields a set of M edge-disjoint qt -paths, so $\lambda_{G_2}(q, t) \geq M$ by Theorem 2.1.2. \square

For some $v \in V_H$, let \mathbf{B}_v denote $(L(W^v) - L_m(W^v)) \cup (P(W^v) - P_m(W^v))$. Now we define \mathbf{F} to be an odd-vertex pairing of G_2 in the following way: For every $uv \in E_H$, \mathbf{F} contains an edge between B_u and B_v . This is possible because for every $v \in V_H$, the set of vertices in $V(W^v)$ which are of odd degree in G_2 is exactly B_v and $|B_v| = d_H(v)$.

3.1.1.3.4 Conclusion

This subsection is dedicated to finishing the reduction. We first take care of the reduction for CA.

Lemma 3.1.6. (G_2, F) is a negative instance of CA if and only if there is some $q\bar{t}$ -set $X \subseteq V_1$ such that $d_{G_1}(X) - d_H(X \cap V_H) < M$.

Proof. First suppose that there is some $q\bar{t}$ -set $X \subseteq V_1$ such that $d_{G_1}(X) - d_H(X \cap V_H) < M$. Let $X' \subseteq V_2$ be the set that contains $q \cup \cup_{v \in X} V(W^v)$ and that contains $V(W^s)$ if X contains s . Then Lemma 3.1.5 yields $d_{G_2}(X') - d_F(X') = d_{G_1}(X) - d_H(X \cap V_H) < M = R_{G_2}(X')$, so (G_2, F) is a negative instance of CA.

Now suppose that (G_2, F) is a negative instance of CA, so there is some $X \subsetneq V_2$ such that $d_{G_2}(X) - d_F(X) < R_{G_2}(X)$. We choose X among all such sets such that $d_{G_2}(X)$ is minimal.

Claim 3.1.2. Let $W \in W^s \cup \{W^v : v \in V_H\}$. Then each connected component of $W[X]$ or $W[V_2 - X]$ contains at least two vertices of $L(W) \cup P(W)$.

Proof. By symmetry and as $d_{G_2}(X) = d_{G_2}(V_2 - X)$, it suffices to prove the statement for $W[X]$. For the sake of a contradiction, suppose that for the vertex set C of a connected component of $W[X]$, we have $|C \cap (L(W) \cup P(W))| \leq 1$.

First suppose that $X = C$. If X consists of a single vertex v with $d_F(v) = 1$, Lemma 3.1.5 yields $d_{G_2}(X) - d_F(X) = 3 - 1 = 2 = R_{G_2}(X)$, a contradiction. Otherwise, Lemma 3.1.2 yields $d_{G_2}(X) \geq 4$ and so, as $d_F(X) \leq 1$ and $G \cup F$ is Eulerian, we obtain by Lemma 3.1.5 that $d_{G_2}(X) - d_F(X) \geq 4 = R_{G_2}(X)$, a contradiction.

We may hence suppose that $X' = X - C$ is nonempty, so, by Lemma 3.1.5 and as $q, t \notin V(W)$, we have $R_{G_2}(X) - R_{G_2}(X') \leq 4 - 2 = 2$. If C consists of a single vertex v with $d_F(v) = 0$, we obtain $d_{G_2}(X') - d_F(X') \leq d_{G_2}(X) - 2 - d_F(X) \leq R_{G_2}(X) - 2 \leq R_{G_2}(X')$, a contradiction to the minimality of X . Otherwise, Lemma 3.1.2 yields $d_{G_2}(X) - d_{G_2}(X') \geq d_W(X) - 1 \geq 4 - 1 = 3$ and $d_F(X') - d_F(X) \leq 1$. This yields $d_{G_2}(X') - d_F(X') \leq (d_{G_2}(X) - 3) - (d_F(X) - 1) \leq R_{G_2}(X) - 2 \leq R_{G_2}(X')$, a contradiction to the minimality of X . □

We are now ready to show that $V(W) \subseteq X$ or $V(W) \cap X \neq \emptyset$ for every $W \in W^s \cup \{W^v : v \in V_H\}$. Suppose otherwise, then by Claim 3.1.2, both $W[X]$ and $W[V(W) - X]$ have a connected component each containing at least two vertices of $L(W) \cup P(W)$. By Lemmas 3.1.3 and 3.1.5, this yields $d_{G_2}(X) - d_F(X) \geq M + m + 1 - m > M \geq R_{G_2}(X)$, a contradiction.

Now let $X^* \subseteq V_1$ be the set of vertices that contains v whenever $V(W^v) \subseteq X$ and s if $V(W^s) \subseteq X$. Observe that $d_{G_2}(X) = d_{G_1}(X^*) \geq 2m$ by construction. Also, observe that $d_F(X) = d_H(X^* \cap V_H)$. By symmetry, we may suppose that $q \in X$. If X is not a $q\bar{t}$ -set, Lemma 3.1.5 yields $d_{G_2}(X) - d_F(X) \geq d_{G_1}(X^*) - m \geq 2m - m = m > 4 \geq R_{G_2}(X)$, a contradiction. If X^* is a $q\bar{t}$ -set, by Lemma 3.1.5, we obtain $d_{G_1}(X^*) - d_H(X^* \cap V_H) = d_{G_2}(X) - d_F(X) < R_{G_2}(X) = M$. □

We next establish a similar result for OA. The following result can be obtained by analogous methods to the proof of Lemma 3.1.6. Several arguments simplify.

Lemma 3.1.7. There is no $X \subseteq V_2$ such that $d_{G_2}(X) < d_F(X)$.

We here prove the following result that allows for a reduction for OA. While this proof does not require any new arguments apart from Lemma 3.1.7, we include it here for the sake of selfcontainment. The second implication can be found in a similar form in [81].

Lemma 3.1.8. (G_2, F) is a negative instance of OA if and only if (G_2, F) is a negative instance of CA.

Proof. If (G_2, F) is a negative instance of OA, it also is a negative instance of CA by Lemma 3.1.1.

For the other direction, suppose that (G_2, F) is a negative instance of CA, so there is some nonempty $X \subsetneq V_2$ such that $d_{G_2}(X) - d_F(X) < R_{G_2}(X)$. Let $u \in X$ and $v \in V_2 - X$ such that $R_{G_2}(X) = 2 \lfloor \frac{\lambda_{G_2}(u,v)}{2} \rfloor$. Let \vec{F} be an orientation of F such that all the edges with exactly one endvertex in X are directed away from X . By Lemma 3.1.7 and Theorem 2.1.4, there is an orientation \vec{G}_2 of G_2 such that $\vec{G}_2 \cup \vec{F}$ is Eulerian. This yields

$$\begin{aligned} \lambda_{\vec{G}_2}(u, v) &\leq d_{\vec{G}_2}^+(X) \\ &= \frac{1}{2}(d_{G_2}(X) + d_F(X)) - d_{\vec{F}}^+(X) \\ &= \frac{1}{2}(d_{G_2}(X) + d_F(X)) - d_F(X) \\ &= \frac{1}{2}(d_{G_2}(X) - d_F(X)) \\ &< \frac{1}{2}R_{G_2}(X) \\ &= \lfloor \frac{\lambda_{G_2}(u, v)}{2} \rfloor. \end{aligned}$$

We obtain that \vec{G}_2 is not well-balanced, so (G_2, F) is a negative instance of OA. \square

We are now ready to give the conclusion for Theorems 3.1.19 and 3.1.20.

By Lemmas 3.1.4 and 3.1.6, we obtain that (G_2, F) is a negative instance of CA if and only if (H, k) is a positive instance of AMAXCUT. By Lemma 2.3.2 and as the size of (G_2, F) is polynomial in the size of (H, k) , we obtain Theorem 3.1.19.

By Lemmas 3.1.4, 3.1.6 and 3.1.8, we obtain that (G_2, F) is a negative instance of OA if and only if (H, k) is a positive instance of AMAXCUT. By Lemma 2.3.2 and as the size of (G_2, F) is polynomial in the size of (H, k) , we obtain Theorem 3.1.20.

3.1.2 Orientations for local arc-connectivity

In this part, we deal with a problem that is much more general than finding a k -arc-connected orientation or a well-balanced orientation. Given a graph $G = (V, E)$ and some requirement function $r : V^2 \rightarrow \mathbb{Z}_{\geq 0}$, we wish to find an orientation \vec{G} of G such that $\lambda_{\vec{G}}(u, v) \geq r(u, v)$ for all $u, v \in V$. We call r *symmetric* if $r(u, v) = r(v, u)$ for all $v \in V$. Observe that k -arc-connected orientations and well-balanced orientations are obtained as special cases when for all $u, v \in V$, we have $r(u, v) = k$ and $r(u, v) = \lfloor \frac{\lambda_G(u,v)}{2} \rfloor$, respectively.

Formally, we consider the following problem:

Local arc-connectivity orientation (LACO):

Instance: A graph G and a requirement function $r : V^2 \rightarrow \mathbb{Z}_{\geq 0}$.

Question: Is there an orientation \vec{G} of G such that $\lambda_{\vec{G}}(u, v) \geq r(u, v)$ for all $u, v \in V$?

The main result of this part is the following.

Theorem 3.1.21. *LACO is NP-complete.*

In Section 3.1.2.1, we prove Theorem 3.1.21. In Section 3.1.2.2, we consider some more restricted cases of LACO.

3.1.2.1 Hardness result

This section is dedicated to proving Theorem 3.1.21. Recall the algorithmic problem BWBO from Section 3.1.1.2. By Theorem 3.1.11, BWBO is NP-complete.

We now give the reduction for Theorem 3.1.21.

Proof. (of Theorem 3.1.21)

We prove this by a reduction from BWBO. Let $(\mathbf{G} = (\mathbf{V}, \mathbf{E}), l^+, l^-)$ be an instance of BWBO. We add two vertices \mathbf{x} and \mathbf{y} and for every $v \in V$, we add $d_G(v)$ edges between v and each of x and y . We denote this graph by $\mathbf{G}' = (\mathbf{V}', \mathbf{E}')$. Observe that $|V'| = |V| + 2$ and $|E'| = 5|E|$, so the size of G' is polynomial in the size of G . We now define $\mathbf{r}: (V')^2 \rightarrow \mathbb{Z}_{\geq 0}$ by $r(u, v) = \lfloor \frac{\lambda_G(u, v)}{2} \rfloor$ for all $u, v \in V$, $r(x, v) = d_G(v) + l^-(v)$, $r(v, x) = 0$, $r(y, v) = 0$ and $r(v, y) = d_G(v) + l^+(v)$ for all $v \in V$, $r(x, y) = 2|E|$ and $r(y, x) = 0$.

We prove that (G', r) is a positive instance of LACO if and only if (G, l^+, l^-) is a positive instance of BWBO. First suppose that (G', r) is a positive instance of LACO, so there is an orientation $\vec{\mathbf{G}}'$ of G' such that $\lambda_{\vec{\mathbf{G}}'}(u, v) \geq r(u, v)$ for all $u, v \in V'$. Observe that $d_{G'}(x) = r(x, y) = d_{G'}(y)$, so x is a root and y is a sink in $\vec{\mathbf{G}}'$. We show that $\vec{\mathbf{G}}$, the restriction of $\vec{\mathbf{G}}'$ to G , is a well-balanced orientation of G such that $d_{\vec{\mathbf{G}}}^+(v) \geq l^+(v)$ and $d_{\vec{\mathbf{G}}}^-(v) \geq l^-(v)$ for all $v \in V$. As x is a root and y is a sink in $\vec{\mathbf{G}}'$, for any $u, v \in V$, we have $\lambda_{\vec{\mathbf{G}}'}(u, v) = \lambda_{\vec{\mathbf{G}}'}(u, v) \geq r(u, v) = \lfloor \frac{\lambda_G(u, v)}{2} \rfloor$, so $\vec{\mathbf{G}}$ is well-balanced. Further, for any $v \in V$, we have $d_{\vec{\mathbf{G}}}^-(v) = d_{\vec{\mathbf{G}}'}^-(v) - d_{\vec{\mathbf{G}}'}^-(x, v) \geq \lambda_{\vec{\mathbf{G}}'}(x, v) - d_{\vec{\mathbf{G}}'}^-(x, v) \geq r(x, v) - d_{\vec{\mathbf{G}}'}^-(x, v) = d_G(v) + l^-(v) - d_G(v) = l^-(v)$. Similarly, $d_{\vec{\mathbf{G}}}^+(v) \geq l^+(v)$, so (G, l^+, l^-) is a positive instance of BWBO.

Now suppose that (G, l^+, l^-) is a positive instance of BWBO, so there is a well-balanced orientation $\vec{\mathbf{G}}$ of G such that $d_{\vec{\mathbf{G}}}^+(v) \geq l^+(v)$ and $d_{\vec{\mathbf{G}}}^-(v) \geq l^-(v)$ for all $v \in V$. We complete this to an orientation $\vec{\mathbf{G}}'$ of G' by orienting all edges incident to x away from x and all edges incident to y toward y . As $\vec{\mathbf{G}}$ is well-balanced, we have $\lambda_{\vec{\mathbf{G}}'}(u, v) = \lambda_{\vec{\mathbf{G}}}(u, v) \geq \lfloor \frac{\lambda_G(u, v)}{2} \rfloor = r(u, v)$ for all $u, v \in V$. By construction, we have $\lambda_{\vec{\mathbf{G}}'}(x, y) = \sum_{v \in V} d_G(v) = 2|E| = r(x, y)$. For any $v \in V$, we have $d_G(v)$ arc-disjoint xv -paths of length 1. Further, for every arc uv entering v in $\vec{\mathbf{G}}$, we have a path xuv . As all these paths can be chosen to be arc-disjoint, we obtain by Theorem 2.1.3 that $\lambda_{\vec{\mathbf{G}}'}(x, v) \geq d_{\vec{\mathbf{G}}'}^-(x, v) + d_{\vec{\mathbf{G}}}^-(v) \geq d_G(v) + l^-(v) = r(x, v)$. Similarly, $\lambda_{\vec{\mathbf{G}}'}(v, y) \geq r(v, y)$, so (G', r) is a positive instance of LACO. \square

3.1.2.2 Restricted cases

Certainly, Theorem 3.1.21 cannot be the last word on the problem of finding orientations of local arc-connectivity. Numerous special cases can be considered. For example, the following is an immediate consequence of Theorem 3.1.6.

Corollary 3.1.1. *An instance (G, r) of LACO where r is symmetric is positive if and only if $r(u, v) \leq \lfloor \frac{\lambda_G(u, v)}{2} \rfloor$ for all $u, v \in V$.*

We give two further results for more restricted cases one of which is negative and one of which is positive. The first one is a simple consequence of Theorem 3.1.6.

Corollary 3.1.2. *LACO is NP-complete even if G is bipartite.*

Proof. We prove this by a reduction from LACO. Let $(\mathbf{G} = (\mathbf{V}, \mathbf{E}), \mathbf{r})$ be an instance of LACO. Let $\mathbf{G}' = (\mathbf{V} \cup \mathbf{V}_E, \mathbf{E}')$ be obtained from G by replacing every edge $e = uv \in E$ by a new vertex \mathbf{x}_e and the edges $u\mathbf{x}_e$ and $v\mathbf{x}_e$. Observe that G' is bipartite. We next define $\mathbf{r}' : V \cup V_E \rightarrow \mathbb{Z}_{\geq 0}$ by $r'(u, v) = r(u, v)$ if $u, v \in V$ and $r(u, v) = 0$ otherwise. We

show that (G, r) is a positive instance of LACO if and only if (G', r') is a positive instance of LACO.

First suppose that (G, r) is a positive instance of LACO, so there is an orientation $\vec{G} = (\mathbf{V}, \vec{E})$ of G such that $\lambda_{\vec{G}}(u, v) \geq r(u, v)$ for all $u, v \in V$. Let \vec{G}' be obtained from \vec{G} by replacing every $\vec{e} = uv \in \vec{E}$ by the vertex x_e and the arcs ux_e and x_ev . Clearly, \vec{G}' is an orientation of G' and we have $\lambda_{\vec{G}'}(u, v) = \lambda_{\vec{G}}(u, v) \geq r(u, v) = r'(u, v)$ for all $u, v \in V$, so (G', r') is a positive instance of LACO.

Now suppose that (G', r') is a positive instance of LACO, so there is an orientation $\vec{G}' = (V \cup V_E, \vec{E}')$ of G such that $\lambda_{\vec{G}'}(u, v) \geq r'(u, v)$ for all $u, v \in V$. We now create an orientation \vec{G} of G in the following way. If for some edge $e = uv \in E$, both the arcs ux_e and x_ev are contained in \vec{E}' , we orient e from u to v . On the other hand, if $d_{\vec{G}'}^-(x_e) \in \{0, 2\}$ for some $e \in E$, we give e an arbitrary orientation. For all $u, v \in V$, we obtain $\lambda_{\vec{G}}(u, v) \geq \lambda_{\vec{G}'}(u, v) \geq r'(u, v) = r(u, v)$, so (G, r) is a positive instance of LACO. \square

We next give a positive result.

Theorem 3.1.22. *There is a polynomial time algorithm for LACO if $r(u, v) \leq 1$ for all $u, v \in V$.*

Proof. Let $(\mathbf{G} = (\mathbf{V}, \mathbf{E}), \mathbf{r})$ be an instance of LACO such that $r(u, v) \leq 1$ for all $u, v \in V$. If G is 2-edge-connected, then there is a strongly connected orientation of G by Theorem 3.1.2, so, in particular, (G, r) is a positive instance of LACO. We may hence suppose that there is some partition (S_1, S_2) of V such that $d_G(S_1, S_2) \leq 1$. For $i = 1, 2$, let $\mathbf{G}_i = (\mathbf{S}_i, \mathbf{E}_i) = G[S_i]$. If $\max\{r(u, v) : u \in S_1, v \in S_2\} + \max\{r(u, v) : u \in S_2, v \in S_1\} > d_G(S_1, S_2)$, we obtain that (G, r) is a negative instance of LACO. Also, if $\max\{r(u, v) : u \in S_1, v \in S_2\} + \max\{r(u, v) : u \in S_2, v \in S_1\} = 0$, then (G, r) is a positive instance of LACO if and only if $(G_i, r[S_i])$ is a positive instance of LACO for $i = 1, 2$. By $1 \leq \max\{r(u, v) : u \in S_1, v \in S_2\} + \max\{r(u, v) : u \in S_2, v \in S_1\} \leq d_G(S_1, S_2) \leq 1$ and by symmetry, it remains to consider the case that $\max\{r(u, v) : u \in S_1, v \in S_2\} = 1, \max\{r(u, v) : u \in S_2, v \in S_1\} = 0$ and $d_G(S_1, S_2) = 1$. Let $\mathbf{x}_1\mathbf{x}_2$ be the edge of G between S_1 and S_2 with $x_i \in S_i$ for $i = 1, 2$. We now define two requirement functions $\mathbf{r}_i : (S_i)^2 \rightarrow \mathbb{Z}_{\geq 0}$. For all $u \in S_1 - x_1$ for which there is a $v \in S_2$ with $r(u, v) = 1$, we define $r_1(u, x_1) = 1$. For all other pairs $u, v \in S_1$, we define $r_1(u, v) = r(u, v)$. For all $v \in S_2 - x_2$ for which there is a $u \in S_1$ with $r(u, v) = 1$, we define $r_2(x_2, v) = 1$. For all other pairs $u, v \in S_2$, we define $r_2(u, v) = r(u, v)$.

We next show that (G, r) is a positive instance of LACO if and only if both (G_1, r_1) and (G_2, r_2) are positive instances of LACO.

First suppose that (G, r) is a positive instance of LACO, so there is an orientation $\vec{G} = (\mathbf{V}, \vec{E})$ of G such that $\lambda_{\vec{G}}(u, v) \geq r(u, v)$ for all $u, v \in V$. Let $\vec{G}_i = \vec{G}[S_i]$ for $i = 1, 2$. Clearly, \vec{G}_i is an orientation of G_i . Further, we have $\lambda_{\vec{G}_i}(u, v) = \lambda_{\vec{G}}(u, v) \geq r(u, v)$ for all $u, v \in S_i$. In order to prove that (G_1, r_1) is a positive instance of LACO, it remains to prove that $\lambda_{\vec{G}_1}(u, x_1) \geq 1$ if there is some $v \in S_2$ such that $r(u, v) = 1$. As $\lambda_{\vec{G}}(u, v) \geq r(u, v) = 1$, there is a path from u to v in \vec{G} . Clearly, this path needs to use the arc x_1x_2 . This yields $\lambda_{\vec{G}_1}(u, x_1) = \lambda_{\vec{G}}(u, x_1) \geq 1$. Similarly, (G_2, r_2) is a positive instance of LACO.

Now suppose that both (G_1, r_1) and (G_2, r_2) are positive instances of LACO, so for $i = 1, 2$, there is an orientation \vec{G}_i of G_i such that $\lambda_{\vec{G}_i}(u, v) \geq r_i(u, v)$ for all $u, v \in S_i$. We create an orientation \vec{G} of G by giving every edge in E_i its orientation in \vec{G}_i for $i = 1, 2$ and orienting the edge between x_1 and x_2 from x_1 to x_2 . Clearly, if $u, v \in S_i$ for some $i = 1, 2$, we have $\lambda_{\vec{G}}(u, v) \geq \lambda_{\vec{G}_i}(u, v) \geq r_i(u, v) = r(u, v)$. By assumption, it remains to consider the case that $u \in S_1, v \in S_2$ and $r(u, v) = 1$. In this case, we have

$\lambda_{\vec{G}}(u, x_1) \geq \lambda_{\vec{G}_1}(u, x_1) \geq r_1(u, x_1)$, so there is a path from u to x_1 in \vec{G} . Similarly, there is a path from x_2 to v in \vec{G} . Together with the arc x_1x_2 this yields a path from u to v . It follows that (G, r) is a positive instance of LACO.

We repeat this operation until all the graphs in consideration are either 2-edge-connected or consist of a single vertex. If we reach this state without having found a negative instance, the entire instance is positive by Theorem 3.1.2. Clearly, the above operation is executed at most $|V|$ times, so the algorithm runs in polynomial time. \square

The following is an easy consequence of Theorem 3.1.22.

Corollary 3.1.3. *There is a polynomial time algorithm for LACO if G is a tree.*

Proof. Let (\mathbf{G}, \mathbf{r}) be an instance of LACO such that G is a tree. Clearly, if $r(u, v) \geq 2$ for some $u, v \in V$, then (G, r) is a negative instance of LACO. Otherwise, Theorem 3.1.22 can be applied. \square

Numerous further special cases can be considered. Firstly, one could consider different restrictions on the graph G .

Research Problem 3.1.3. *Find further graph classes \mathcal{G} such that there is a polynomial time algorithm for LACO if $G \in \mathcal{G}$.*

One particularly interesting such class is complete graphs, in other words the question whether we can find a tournament with prescribed arc-connectivities in polynomial time. Further, planar graphs or Eulerian graphs could be considered.

One could also consider restrictions on the function r . The following question is motivated by Theorem 3.1.22.

Research Problem 3.1.4. *Given some $k \in \mathbb{Z}_{\geq 0}$, what is the complexity of LACO if $r(u, v) \leq k$ for all $u, v \in V$?*

This question is open even for $k = 2$.

Another case to consider is when the requirement function is positive only on a small number of vertices.

Research Problem 3.1.5. *Given some $k \in \mathbb{Z}_{\geq 0}$, what is the complexity of LACO if $r(u, v) = 0$ for all $(u, v) \in V^2 - S^2$ for some $S \subseteq V$ with $|S| \leq k$?*

Finally, the following problem has been brought to our attention by Frank [45]. It concerns requirement functions which are close to being symmetric.

Research Problem 3.1.6. *What is the complexity of LACO if $|r(u, v) - r(v, u)| \leq 1$ for all $u, v \in V$?*

3.1.3 Orientations for arc-connectivity and parity

This section is dedicated to adding a new aspect to the theory of graph orientations for arc-connectivity: parity. Through Nash-Williams' odd-vertex pairing theorem (Theorem 3.1.7), we have already encountered an occasion where parity considerations in the given graph led to an important orientation theorem. The use of T -joins in Section 3.1.4 will be another example. In this chapter, we deal with a different relation between orientations, connectivity and parity. Given an undirected graph, we wish to find an orientation that satisfies some connectivity properties and such that the indegree of the vertices of the graph is of a given parity. More precisely, given a graph $G = (V, E)$ and some $T \subseteq V$, we say that an orientation \vec{G} of G is T -odd if $d_{\vec{G}}^-(v)$ is odd for all $v \in T$ and $d_{\vec{G}}^-(v)$ is even

for all $v \in V - T$. For two integers a, b , we use $\mathbf{a} \equiv \mathbf{b}$ to denote that a and b are of the same parity. First observe that the problem becomes easy when we drop the connectivity condition. The following simple observation is well-known.

Proposition 3.1.1. *Let $G = (V, E)$ be a connected graph and $T \subseteq V$. Then G has a T -odd orientation if and only if $|T| \equiv |E|$.*

Proof. If G has a T -odd orientation \vec{G} , we have $|T| \equiv \sum_{v \in V} d_{\vec{G}}^-(v) = |E|$.

To see the other direction, let G' be obtained from G by replacing every edge $e = uv$ by a new vertex x_e and the edges ux_e and vx_e . Clearly, a T -odd orientation of G corresponds to a T' -join in G' where T' is obtained from T by adding all the new vertices. Observe that $|T'| = |T| + |E| \equiv 0$, so the T' -join exists by Proposition 2.1.8. \square

The following observation will be used several times in this chapter.

Proposition 3.1.2. *Let $G = (V, E)$ be a graph, $T \subseteq V$ such that $|T| \equiv |E|$ and $x \in V$. Let \vec{G} be an orientation of G such that $d_{\vec{G}}^-(v) \equiv |T \cap v|$ for all $v \in V - x$. Then \vec{G} is T -odd.*

Proof. We only need to show that $d_{\vec{G}}^-(x) \equiv |T \cap x|$. This follows from $d_{\vec{G}}^-(x) = |E| - \sum_{v \in V-x} d_{\vec{G}}^-(v) \equiv |T| + |T \cap (V - x)| \equiv |T \cap x|$. \square

In [48], Frank, Jordán and Szigeti consider the problem of combining parity and root-connectivity in orientations of undirected graphs. They prove the following result.

Theorem 3.1.23. *Let $G = (V \cup r, E)$ be a singularly rooted graph, $T \subseteq V \cup r$ and k a positive integer. Then there is a T -odd k -root-connected orientation of G if and only if*

$$e_G(\mathcal{P}) \geq kt + s(\mathcal{P})$$

holds for every subpartition $\mathcal{P} = \{X_1, \dots, X_t\}$ of V where $s(\mathcal{P})$ denotes the number of classes X of \mathcal{P} such that $i_G(X) + |T \cap X| + k$ is odd.

Unfortunately, there is no clear way to obtain an algorithm for the corresponding decision problem from the proof of Theorem 3.1.23. The main focus of this chapter will be on the following problem that is probably the most interesting one among several related problems.

Parity constrained strongly connected orientation (PSCO)

Instance: A graph $G = (V, E)$ and $T \subseteq V$.

Question: Is there a T -odd strongly connected orientation of G ?

We are interested in this question both from a structural and algorithmic point of view. Nevertheless, we state all our results in the algorithmic form and wish to determine whether PSCO is algorithmically tractable. We hope for a possible algorithm to also imply a characterization. Unfortunately, in [80], Király and Szabó prove that for PSCO we cannot hope for a characterization which is similar to the one in Theorem 3.1.23. In an attempt to resolve the algorithmic tractability of PSCO, Frank and Király find a characterization for a problem closely related to PSCO [49]. This yields the following algorithmic result:

Theorem 3.1.24. *Given a graph $G = (V, E)$, we can decide in polynomial time whether for all $T \subseteq V$ with $|T| \equiv |E|$, there is a T -odd strongly connected orientation of G .*

This chapter is organized as follows. In Section 3.1.3.1, we give some ideas for finding a polynomial time algorithm for PSCO. More concretely, we show that PSCO is equivalent to a seemingly less general problem. In Section 3.1.3.2, we try to approach the problem from the other direction, meaning we give some problems that are harder or seem to be harder than PSCO. Hardness results for these problems can be considered intermediate steps for proving that PSCO is not algorithmically tractable. We show that one of these problems is hard indeed.

3.1.3.1 Positive approaches

In this section, we consider a problem which is seemingly much less general than PSCO. In particular, it is restricted to graphs all of whose vertices have small degree. We show that this problem is algorithmically equivalent to PSCO. More concretely, we consider the following problem:

Restricted Parity constrained Strongly Connected Orientation(**RPSCO**)

Instance: A 3-edge-connected, 2-vertex-connected graph $G = (V, E)$ such that $d_G(v) \in \{3, 4\}$ for all $v \in V$.

Question: Is there a V -odd strongly connected orientation of G ?

The following result is the main contribution of this section.

Theorem 3.1.25. *There is a polynomial time algorithm for PSCO if and only if there is a polynomial time algorithm for RPSCO.*

We first collect some preliminaries for the proof of Theorem 3.1.25. Given a graph $G = (V, E)$ the operation of *detaching* some $v \in V$ consists of replacing v by two vertices v_1 and v_2 and adding one of the edges uv_1 and uv_2 for every $uv \in E$. The detachment is called *equitable* if the degrees of v_1 and v_2 differ by at most 1 in the new graph.

Lemma 3.1.9. *Let $(G = (V, E), T)$ be an instance of PSCO such that G is 3-edge-connected and 2-vertex-connected and $x \in V$. Let $G' = (V', E')$ be obtained from G by equitably detaching x into two vertices y and z and adding a pair of parallel edges between y and z . Further, let $T' \subseteq V'$ satisfy $T' - \{y, z\} = T - x$ and $|T' \cap \{y, z\}| \equiv |T \cap x|$. Then the following hold:*

- (i) G' is 3-edge-connected and 2-vertex-connected,
- (ii) (G', T') is a positive instance of PSCO if and only if (G, T) is a positive instance of PSCO.

Proof. (i): Let S be a nonempty, proper subset of V' . If S does not separate $\{y, z\}$, without loss of generality $S \cap \{y, z\} = \emptyset$, as G is 3-edge-connected, we obtain $d_{G'}(S) = d_G(S) \geq 3$. If S separates $\{x, y\}$, without loss of generality $x \in S$ and $y \in V' - S$, we obtain from the 2-vertex-connectivity of G that $d_{G'}(S) \geq d_{G'}(y, z) + d_{G' - \{y, z\}}(S - y, V - S - z) = 2 + d_{G-x}(S - y, V - S - z) \geq 2 + 1 = 3$, so G' is 3-edge-connected.

For some $p \in V' - \{y, z\}$, as $G - p$ and $G'[\{x, y\}]$ are connected, so is $G' - p$. Further, we have that $G' - \{y, z\} = G - x$ is connected. As the detachment is equitable and G is 3-edge-connected, we further obtain $d_{G'-z}(y) \geq \lfloor \frac{d_G(x)}{2} \rfloor \geq 1$, so $G' - z$ is connected. Similarly, $G' - y$ is connected and so G' is 2-vertex-connected.

(ii): First suppose that (G', T') is a positive instance of PSCO, so there is a T' -odd strongly connected orientation \vec{G}' of G' . Let \vec{G} be obtained from \vec{G}' by contracting $\{y, z\}$ into x . By Proposition 2.1.2, \vec{G} is strongly connected. Further, observe that

$|T'| \equiv |T| \equiv |E| \equiv |E'|$ and $d_{\vec{G}}^-(v) = d_{G'}^-(v) \equiv |T' \cap v| = |T \cap v|$ for all $v \in V - x$. It now follows from Proposition 3.1.2 that \vec{G} is T -odd.

Now suppose that (G, T) is a positive instance of $PSCO$, so there is a T -odd strongly connected orientation \vec{G} of G . We now define three orientations of G' . First, in \vec{G} , we detach x into y and z in the same way as we did when creating G' from G . Then, we create \vec{G}'_0 by adding one arc from y to z and one arc from z to y , we create \vec{G}'_1 by adding two arcs from y to z and we create \vec{G}'_2 by adding two arcs from z to y . Observe that \vec{G}'_0, \vec{G}'_1 and \vec{G}'_2 are orientations of G' . Clearly, as \vec{G} is strongly connected, so is \vec{G}'_0 . If \vec{G}'_0 is T' -odd, there is nothing to prove, so suppose otherwise. As $d_{\vec{G}'_0}^-(v) = d_{\vec{G}}^-(v) \equiv |T \cap v| = |T' \cap v|$ for all $v \in V - \{y, z\}$, we may suppose by symmetry that $d_{\vec{G}'_0}^-(y) \not\equiv |T' \cap y|$. As $d_{\vec{G}'_1}^-(v) = d_{\vec{G}'_0}^-(v) \equiv |T' \cap v|$ for all $v \in V - \{y, z\}$ and $d_{\vec{G}'_1}^-(y) \equiv d_{\vec{G}'_0}^-(y) + 1 \equiv |T' \cap y|$, we obtain by $|T'| \equiv |E'|$ and Proposition 3.1.2 that \vec{G}'_1 is T' -odd. Similarly, \vec{G}'_2 is T' -odd. We will finish the proof by showing that one of \vec{G}'_1 and \vec{G}'_2 is strongly connected.

Suppose otherwise, so there are nonempty sets $\mathbf{S}_1, \mathbf{S}_2 \subsetneq V'$ such that $d_{\vec{G}'_i}^-(S_i) = 0$ for $i = 1, 2$. As \vec{G} is strongly connected and by construction, we obtain that $y \in S_1, z \in V' - S_1, y \in V' - S_2$ and $z \in S_2$. This yields, by Proposition 2.2.3,

$$\begin{aligned} 0 &\leq d_{\vec{G}'_0}^-(S_1 \cup S_2) + d_{\vec{G}'_0}^-(S_1 \cap S_2) \\ &= d_{\vec{G}'_0}^-(S_1) + d_{\vec{G}'_0}^-(S_2) - d_{G'}(S_1, S_2) \\ &\leq d_{\vec{G}'_1}^-(S_1) + 1 + d_{\vec{G}'_2}^-(S_2) + 1 - d_{G'}(y, z) - d_{G'}(S_1 - y, S_2 - z) \\ &= 0 + 1 + 0 + 1 - 2 - d_G(S_1 - y, S_2 - z) \\ &= -d_G(S_1 - y, S_2 - z) \\ &\leq 0, \end{aligned}$$

so equality holds throughout. As \vec{G}'_0 is strongly connected, we obtain that $S_1 \cup S_2 = V'$ and $S_1 \cap S_2 = \emptyset$. This yields that $\{S_1 - y, S_2 - z\}$ is a partition of $V' - \{y, z\} = V - x$. As $d_G(S_1 - y, S_2 - z) = 0$, we obtain that x is a cutvertex of G , a contradiction to G being 2-vertex-connected. \square

We are now ready to start with the main proof of Theorem 3.1.25.

Proof. (of Theorem 3.1.25) Clearly, RPSCO is a special case of PSCO, so a polynomial time algorithm for PSCO implies a polynomial time algorithm for RPSCO. We now suppose that there is a polynomial time algorithm for RPSCO and show through several claims that there is a polynomial time algorithm for PSCO.

Claim 3.1.3. *There is a polynomial time algorithm that decides whether an instance $(G = (V, E), V)$ of PSCO is positive if G is 3-edge-connected and 2-vertex-connected and $d_G(v) \in \{3, 4, 5\}$ for all $v \in V$.*

Proof. Let $(\mathbf{G} = (\mathbf{V}, \mathbf{E}), \mathbf{V})$ be an instance of PSCO such that G is 3-edge-connected and 2-vertex-connected and $d_G(v) \in \{3, 4, 5\}$ for all $v \in V$. By Proposition 3.1.1, we may suppose that $|E| \equiv |V|$. If $d_G(v) \in \{3, 4\}$ for all $v \in V$, the desired algorithm exists by assumption. We may hence suppose that there is some $\mathbf{x} \in V$ that is incident to exactly 5 edges $\mathbf{e}_1, \dots, \mathbf{e}_5$. We now create a graph $\mathbf{G}' = (\mathbf{V}', \mathbf{E}')$ by replacing x by 5 vertices $\{\mathbf{x}_1, \dots, \mathbf{x}_5\}$ such that x_i is incident to e_i for $i = 1, \dots, 5$. Further, we add the edge set of a cycle $\mathbf{C} = x_1, \dots, x_5, x_1$ and a vertex \mathbf{z} that is adjacent to the vertices in $\{x_2, \dots, x_5\}$. Let $\mathbf{X} = \{x_1, \dots, x_5\} \cup \mathbf{z}$. An illustration can be found in Figure 3.1.5.

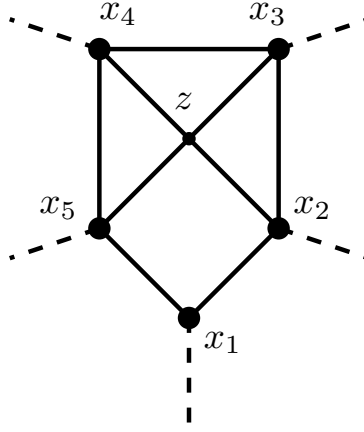


Figure 3.1.5: An illustration for the gadget x is replaced by in Claim 3.1.3. The solid edges are new while the dashed ones exist in G .

Observe that $|E'| = |E| + 9 \equiv |V| + 9 \equiv |V| + 5 = |V'|$. Clearly, $d_{G'}(v) \in \{3, 4, 5\}$ for all $v \in V'$.

We next show that G' is 3-edge-connected. Let S be a nonempty, proper subset of V' . If S does not separate X , say $S \cap X = \emptyset$, we obtain that $d_{G'}(S) = d_G(S) \geq 3$ as G is 3-edge-connected. An easy case analysis shows that $d_G(S) \geq 3$ if $S \subseteq X$. We may hence suppose that S and X are properly intersecting. As $G'[X]$ is 2-edge-connected and G is 2-vertex-connected, this yields $d_{G'}(S) \geq d_{G'[X]}(S \cap X) + d_{G'-X}(S - X, V - S - X) \geq 2 + d_{G-x}(S - X, V - S - X) \geq 2 + 1 = 3$, so G' is 3-edge-connected.

We next show that G' is 2-vertex-connected. For any $p \in V' - X$, observe that $G - p$ is connected and $G'[X]$ is connected, so $G' - p$ is connected. For any $p \in X$, observe that $G - x$ is connected, $G[X] - p$ is connected and there are several vertices in X that are incident to an edge in $\delta_{G'}(X)$, so $G' - p$ is connected. It follows that G' is 2-vertex-connected.

We next show that (G', V') is a positive instance of PSCO if and only if (G, V) is a positive instance of PSCO. First suppose that (G', V') is a positive instance of PSCO, so there is a V' -odd strongly connected orientation \vec{G}' of G' . Let \vec{G} be obtained from \vec{G}' by contracting X into x . By Proposition 2.1.2, \vec{G} is strongly connected. Further, we have $d_{\vec{G}}^-(v) = d_{\vec{G}'}^-(v) \equiv 1$ for all $v \in V - x$. As $|E| \equiv |V|$ and by Proposition 3.1.2, we obtain that \vec{G} is V -odd.

Now suppose that (G, V) is a positive instance of PSCO, so there is a V -odd strongly connected orientation \vec{G} of G . Possibly redirecting a circuit, we may suppose that \vec{e}_1 is directed away from x in \vec{G} . We extend \vec{G} to an orientation \vec{G}' of G' . First, we give all edges in E the same orientation they have in \vec{G} and orient C as a circuit. For $i = 2, \dots, 5$, let $x_i z$ be oriented from x_i to z if \vec{e}_i leaves x_i and from z to x_i if \vec{e}_i enters x_i . By construction, as $|V'| \equiv |E'|$ and by Proposition 3.1.2, we obtain that \vec{G}' is V' -odd. Further, as \vec{G} and \vec{C} are strongly connected, so is $\vec{G}' - z$ by Proposition 2.1.3(b). Finally, as \vec{G}' is V' -odd and $d_{G'}(z) = 4$, we have $d_{\vec{G}'}^-(z), d_{\vec{G}'}^+(z) \geq 1$, so \vec{G}' is strongly connected.

We repeat the above operation as long as H contains a vertex of degree 5 where \mathbf{H} is the current graph at some stage of the algorithm. Let $(\mathbf{G}^*, \mathbf{V}^*)$ be the instance of PSCO obtained after the last application of this operation. Observe that every time the operation is applied, the number of vertices of degree 5 in H decreases by one. It follows that the operation is applied at most $|V|$ times. Further, $|V(\mathbf{H})|$ increases by 5 and $|E(\mathbf{H})|$ increases by 9 each time the operation is applied. It follows that the size of \mathbf{G}^* is polynomial in the size of G and that \mathbf{G}^* can be constructed from G in polynomial time. Finally, by assumption, we can check whether $(\mathbf{G}^*, \mathbf{V}^*)$ is a positive instance of PSCO in

polynomial time, so the statement follows. \square

Claim 3.1.4. *There is a polynomial time algorithm that decides whether an instance $(G = (V, E), T)$ of PSCO is positive if G is 3-edge-connected and 2-vertex-connected and $d_G(v) \in \{3, 4, 5\}$ for all $v \in V$.*

Proof. Let $(\mathbf{G} = (\mathbf{V}, \mathbf{E}), \mathbf{T})$ be an instance of PSCO such that G is 3-edge-connected and 2-vertex-connected and $d_G(v) \in \{3, 4, 5\}$ for all $v \in V$. If $T = V$, there is nothing to prove by Claim 3.1.3, so suppose that there is some $x \in V - T$. Let $\mathbf{G}' = (\mathbf{V}', \mathbf{E}')$ be obtained by equitably detaching x into two vertices y and z and adding two edges between y and z and let $\mathbf{T}' = T \cup \{y, z\}$. Clearly, we have $d_{\mathbf{G}'}(v) \in \{3, 4, 5\}$ for all $v \in V'$. Further, by Lemma 3.1.9(i), \mathbf{G}' is 3-edge-connected and 2-vertex-connected and by Lemma 3.1.9(ii), we have that $(\mathbf{G}', \mathbf{T}')$ is a positive instance of PSCO if and only if (G, T) is a positive instance of PSCO.

We apply this operation as long as $V(H) - T \neq \emptyset$ where \mathbf{H} is the current graph at some stage of the algorithm. Let $(\mathbf{G}^*, \mathbf{V}^*)$ be the instance of PSCO obtained after the last application of this operation. Observe that every time the operation is applied, $|V(H) - T|$ decreases by one. It follows that the operation is applied at most $|V|$ times. Further, $|V(H)|$ increases by 1 and $|E(H)|$ increases by 2 each time the operation is applied. It follows that the size of \mathbf{G}^* is polynomial in the size of G and that \mathbf{G}^* can be constructed from (G, T) in polynomial time. Finally, by Claim 3.1.3, we can check whether $(\mathbf{G}^*, \mathbf{V}^*)$ is a positive instance of PSCO in polynomial time, so the statement follows. \square

Claim 3.1.5. *There is a polynomial time algorithm that decides whether an instance $(G = (V, E), T)$ of PSCO is positive if G is 3-edge-connected and 2-vertex-connected.*

Proof. Let $(\mathbf{G} = (\mathbf{V}, \mathbf{E}), \mathbf{T})$ be an instance of PSCO such that G is 3-edge-connected and 2-vertex-connected. If $d_G(v) \leq 5$ for all $v \in V$, there is nothing to prove by Claim 3.1.4, so suppose that there is some $x \in V$ with $d_G(x) \geq 6$. Let $\mathbf{G}' = (\mathbf{V}', \mathbf{E}')$ be obtained by equitably detaching x into two vertices y and z and adding two edges between y and z and choose $\mathbf{T}' \subseteq V'$ such that $T' - \{y, z\} = T - x$ and $|T' \cap \{y, z\}| \equiv |T \cap x|$. By Lemma 3.1.9 (i), \mathbf{G}' is 3-edge-connected and 2-vertex-connected and by Lemma 3.1.9 (ii), we have that $(\mathbf{G}', \mathbf{T}')$ is a positive instance of PSCO if and only if (G, T) is a positive instance of PSCO.

We apply this operation as long as $d_H(v) \geq 6$ for some $v \in V(H)$ where \mathbf{H} is the current graph at some stage of the algorithm. At some intermediate step of the algorithm, let $M = \sum_{v \in V(H)} \max\{d_H(v) - 5, 0\}$. Observe that every time the above operation is applied, we have $d_G(x), d_{\mathbf{G}'}(y), d_{\mathbf{G}'}(z) \geq 5$ and $(d_{\mathbf{G}'}(y) - 5) + (d_{\mathbf{G}'}(z) - 5) = (d_G(x) - 5) - 1$, so M decreases by 1. Further, before the first application of the operation, we have $M \leq \sum_{v \in V} d_G(v) = 2|E|$ and after the last application of the operation, we have $M \geq 0$. It follows that the operation is applied at most $2|E|$ times. Let $(\mathbf{G}^*, \mathbf{T}^*)$ be the instance of PSCO obtained after the last application of this operation. As the operation is applied at most $2|E|$ times, $(\mathbf{G}^*, \mathbf{T}^*)$ is well-defined, the size of \mathbf{G}^* is polynomial in the size of G and $(\mathbf{G}^*, \mathbf{T}^*)$ can be constructed in polynomial time. Further, $(\mathbf{G}^*, \mathbf{T}^*)$ is a positive instance of PSCO if and only if (G, T) is a positive instance of PSCO. By Claim 3.1.4, we can check in polynomial time whether $(\mathbf{G}^*, \mathbf{T}^*)$ is a positive instance of PSCO. \square

Claim 3.1.6. *There is a polynomial time algorithm that decides whether an instance $(G = (V, E), T)$ of PSCO is positive if G is 3-edge-connected.*

Proof. Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ be a 3-edge-connected graph. By Proposition 3.1.1, we may suppose that $|E| \equiv |T|$. If G is 2-vertex-connected, there is nothing to prove by Claim

3.1.5. Also, if $|V| \leq 2$, a decision algorithm is trivially available. We may hence suppose that there is a partition $\{\mathbf{S}_1, \mathbf{x}, \mathbf{S}_2\}$ of V such that x is a single vertex and $d_G(S_1, S_2) = 0$. For $i = 1, 2$, let $(\mathbf{V}_i, \mathbf{E}_i) = \mathbf{G}_i = G[S_i \cup x]$ and let \mathbf{T}_i be defined in the unique way that $T_i - x = T \cap S_i$ and $|E_i| \equiv |T_i|$. Clearly, for any nonempty $\mathbf{R} \subseteq S_i$, we have $d_{G_i}(\mathbf{R}) = d_G(\mathbf{R}) \geq 3$, so G_1 and G_2 are 3-edge-connected. We next show that (G, T) is a positive instance of PSCO if and only if both (G_1, T_1) and (G_2, T_2) are positive instances of PSCO.

First suppose that (G, T) is a positive instance of PSCO, so there is a T -odd strongly connected orientation \vec{G} of G . For $i = 1, 2$, let $\vec{G}_i = \vec{G}[S_i \cup x]$. Clearly, \vec{G}_i is a strongly connected orientation of G_i . Further, for all $v \in S_i$, we have $d_{\vec{G}_i}^-(v) = d_{\vec{G}}^-(v) \equiv |T \cap v| = |T_i \cap v|$. By $|E_i| \equiv |T_i|$ and Proposition 3.1.2, \vec{G}_i is T_i -odd.

Now suppose that both (G_1, T_1) and (G_2, T_2) are positive instances of PSCO, so there are strongly connected orientations \vec{G}_1 and \vec{G}_2 of G_1 and G_2 . Let \vec{G} be obtained from $\vec{G}_1 \cup \vec{G}_2$ by contracting the two copies of x . Clearly, \vec{G} is a strongly connected orientation of G . Further, for all $i = 1, 2$ and $v \in S_i$, we have $d_{\vec{G}}^-(v) = d_{\vec{G}_i}^-(v) \equiv |T_i \cap v| = |T \cap v|$. By $|E| \equiv |T|$ and Proposition 3.1.2, \vec{G} is T -odd.

We repeat this operation as long as H contains a connected component that is not 2-vertex-connected and has at least 3 vertices where \mathbf{H} is the current graph at some stage of the algorithm. At any point of the algorithm, let \mathbf{C} be the collection of connected components of H and let $\mathbf{M} = \sum_{C \in \mathbf{C}} (|V(C)| - 2)$. Observe that M decreases by 2 every time the operation is applied. Further, we have $M = |V| - 2$ before the first application of the operation and $M \geq 0$ at any time of the algorithm. Let $(\mathbf{G}^*, \mathbf{T}^*)$ be the graph obtained after the last application of the operation. By the above, (G^*, T^*) is well-defined, the size of G^* is polynomial in the size of G and (G^*, T^*) can be computed in polynomial time from (G, T) . Further, (G, T) is a positive instance of PSCO if and only if for every connected component $\mathbf{G}_C^* = (\mathbf{V}_C^*, \mathbf{E}_C^*)$ of G^* , we have that $(G_C^*, T^* \cap V_C^*)$ is a positive instance of PSCO. By Claim 3.1.5 and as a trivial algorithm for graphs of at most 2 vertices is available, we can decide in polynomial time whether for every connected component $G_C^* = (V_C^*, E_C^*)$ of G^* , we have that $(G_C^*, T^* \cap V_C^*)$ is a positive instance of PSCO. \square

Claim 3.1.7. *There is a polynomial time algorithm that decides whether an instance $(G = (V, E), T)$ of PSCO is positive.*

Proof. Let $(\mathbf{G} = (\mathbf{V}, \mathbf{E}), \mathbf{T})$ be an instance of PSCO. By Proposition 3.1.1, we may suppose that $|E| \equiv |T|$. If G is 3-edge-connected, there is nothing to prove by Claim 3.1.6. We may hence suppose that there is a partition $(\mathbf{S}_1, \mathbf{S}_2)$ of V such that $d_G(S_1, S_2) \leq 2$. If $d_G(S_1, S_2) \leq 1$, then (G, T) is a negative instance of PSCO by Theorem 3.1.2, we may hence suppose that $d_G(S_1, S_2) = 2$. Let $\mathbf{x}_1\mathbf{x}_2$ and $\mathbf{y}_1\mathbf{y}_2$ be the two edges in $\delta_G(S)$ where $x_1, y_1 \in S_1$ and $x_2, y_2 \in S_2$. If $|T \cap S_i| \equiv i_G(S_i)$ for some $i = 1, 2$, then in any T -odd orientation x_1x_2 and y_1y_2 are both oriented toward S_1 or both oriented toward S_2 while in any strongly connected orientation, one of them is oriented toward S_1 and one of them is oriented toward S_2 . Hence (G, T) is a negative instance of PSCO. We may therefore suppose that $|T \cap S_i| \equiv i_G(S_i) + 1$ for $i = 1, 2$.

For $i = 1, 2$, let \mathbf{G}_i be obtained from $G[S_i]$ by adding an edge between x_i and y_i and let $\mathbf{T}_i = T \cap S_i$. We show that (G, T) is a positive instance of PSCO if and only if both (G_1, T_1) and (G_2, T_2) are positive instances of PSCO.

First suppose that (G, T) is a positive instance of PSCO, so there is a T -odd strongly connected orientation \vec{G} of G . Clearly, exactly one of the two edges between S_1 and S_2 is oriented toward S_1 . By symmetry, we may suppose that x_1x_2 is oriented toward x_2 and y_1y_2 is oriented toward y_1 . Let $\vec{G}_1 = \vec{G}[S_1] \cup x_1y_1$. Observe that \vec{G}_1 is an orientation of G_1 . As \vec{G}_1 can be obtained from \vec{G} by first contracting S_2 and then suppressing the

new vertex, we obtain by Proposition 2.1.3 that \vec{G}_1 is strongly connected. Further, for all $v \in S_1$, we have $d_{\vec{G}_1}^-(v) = d_{\vec{G}}^-(v) \equiv |T \cap v| = |T_1 \cap v|$, so \vec{G}_1 is T_1 -odd. It follows that (G_1, T_1) is a positive instance of PSCO. Similarly, (G_2, T_2) is a positive instance of PSCO.

Now suppose that for $i = 1, 2$, (G_i, T_i) is a positive instance of PSCO, so there is a T_i -odd strongly connected orientation \vec{G}_i of G_i . By symmetry, we may suppose that in \vec{G}_1 , the edge between x_1 and x_2 is oriented from x_1 to x_2 . Possibly redirecting a circuit, we may suppose that in \vec{G}_2 , the edge between y_2 and x_2 is oriented from y_2 to x_2 . We now create an orientation \vec{G} of G by giving every edge in $E[S_i] - \{x_i y_i\}$ its orientation in \vec{G}_i for $i = 1, 2$ and adding the arcs $x_1 x_2$ and $y_2 y_1$. First observe that for $i = 1, 2$ and $v \in S_i$, we have $d_{\vec{G}}^-(v) = d_{\vec{G}_i}^-(v) \equiv |T_i \cap v| = |T \cap v|$, so \vec{G} is T -odd. For the strong connectivity of \vec{G} , first observe that there is a path from v to x_1 for every $v \in S_1$ in \vec{G}_1 and none of these paths uses the arc $x_1 y_1$, so it still exists in \vec{G} . Similarly, \vec{G} contains a path from v to y_2 for all $v \in S_2$. As there is an arc from y_2 to y_1 and a path from y_1 to x_1 in \vec{G} , we obtain that there is a path from v to x_1 for all $v \in V$. Similarly, there is a path from x_2 to v for all $v \in V$. As there is an arc from x_1 to x_2 in \vec{G} , we obtain that \vec{G} is strongly connected.

We repeat this operation as long as H contains a connected component that is not 3-edge-connected where H is the current graph at some stage of the algorithm. At any point of the algorithm, let \mathcal{C} be the collection of connected components of G and let $M = \sum_{C \in \mathcal{C}} (|V(C)| - 1)$. Observe that M decreases by 1 every time the operation is applied. Further, $M = |V| - 1$ before the first application of the operation and $M \geq 0$ at any time of the algorithm. Let (G^*, T^*) be the graph obtained after the last application of the operation. By the above, (G^*, T^*) is well-defined, the size of G^* is polynomial in the size of G and (G^*, T^*) can be computed in polynomial time from (G, T) . Further, (G, T) is a positive instance of PSCO if and only if for every connected component $G_C^* = (V_C^*, E_C^*)$ of G^* , we have that $(G_C^*, T^* \cap V_C^*)$ is a positive instance of PSCO. By Claim 3.1.5, we can decide in polynomial time whether for every connected component $G_C^* = (V_C^*, E_C^*)$ of G^* , we have that $(G_C^*, T^* \cap V_C^*)$ is a positive instance of PSCO. \square

The combination of the above claims yields the theorem. \square

In the light of Theorem 3.1.25, it seems natural to study classes of graphs with even stronger restrictions on the vertex degrees. One such class is regular graphs. For cubic graphs, a simple characterization is available.

Theorem 3.1.26. *Let $(G = (V, E), T)$ be an instance of PSCO such that G is cubic. Then (G, T) is a positive instance of PSCO if and only if $|T| = \frac{1}{2}|V|$ and $i_G(X) \leq |X \cap T| + 2|X - T| - 1$ for every nonempty $X \subsetneq V$.*

Proof. First suppose that (G, T) is a positive instance of PSCO, so there is a T -odd strongly connected orientation \vec{G} of G . Observe that, as \vec{G} is strongly connected and T -odd, we have $d_{\vec{G}}^-(v) = 1$ for all $v \in T$ and $d_{\vec{G}}^-(v) = 2$ for all $v \in V - T$. This yields $\frac{3}{2}|V| = |E| = \sum_{v \in V} d_{\vec{G}}^-(v) = 2|V| - |T|$, so $|T| = \frac{1}{2}|V|$. Further, for any nonempty $X \subsetneq V$, since \vec{G} is strongly connected, we have $i_G(X) = \sum_{v \in X} d_{\vec{G}}^-(v) - d_{\vec{G}}^-(X) \leq |X \cap T| + 2|X - T| - 1$.

Now suppose that $|T| = \frac{1}{2}|V|$ and $i_G(X) \leq |X \cap T| + 2|X - T| - 1$ for every nonempty $X \subsetneq V$. By Theorem 2.1.1, there is an orientation \vec{G} of G such that $d_{\vec{G}}^-(v) = 1$ for all $v \in T$ and $d_{\vec{G}}^-(v) = 2$ for all $v \in V - T$. Clearly, \vec{G} is T -odd. Further, for any nonempty $X \subsetneq V$, we have $d_{\vec{G}}^-(X) = \sum_{v \in X} d_{\vec{G}}^-(v) - i_G(X) \geq |X \cap T| + 2|X - T| - (|X \cap T| + 2|X - T| - 1) = 1$, so \vec{G} is strongly connected. \square

The next case to consider is 4-regular graphs. Let $(G = (V, E), T)$ be an instance of PSCO such that G is 4-regular. We say that a cut $\delta_G(X)$ for some nonempty $X \subseteq V$ is

bad if $d_G(X) = 2$ and $|T \cap X| \equiv i_G(X)$. Clearly, if (G, T) contains a bad cut, then (G, T) is a negative instance of PSCO. It is an immediate corollary of Theorem 4.1 in [49] that this is the only nontrivial obstacle for 4-regular graphs.

Theorem 3.1.27. *Let (G, T) be an instance of PSCO such that G is 4-regular, $|T| \equiv |E|$ and (G, T) does not contain a bad cut. Then (G, T) is positive.*

3.1.3.2 Negative approaches

This part approaches the complexity of PSCO from the other direction, giving a collection of problems which might be useful as an intermediate step when trying to prove that PSCO is NP-complete. Firstly, we show that combining connectivity and parity in orientations of mixed graphs leads to hard problems, even when considering rooted connectivity instead of strong connectivity. After, we give a collection of problems which are harder or seemingly harder than PSCO and whose algorithmic tractability is open.

3.1.3.2.1 Mixed graphs

This subsection is dedicated to showing that combining parity and connectivity in orientations of mixed graphs is hard, even when considering rooted connectivity. More concretely, we consider the following problem for some $k \geq 1$:

Mixed parity-constrained k -root-connected orientation(MPkRCO)

Instance: A singularly rooted mixed graph $G = (V \cup r, A \cup E)$ and $T \subseteq V$.

Question: Is there a T -odd orientation of G which is k -root-connected from r ?

Theorem 3.1.28. *MPkRCO is NP-complete for all $k \geq 1$.*

We show this by a reduction from (3,B2)-SAT.

Construction Suppose we are given an instance $F = (X, \mathcal{C})$ of (3,B2)-SAT and some $k \geq 1$. Observe that $3|\mathcal{C}| = 4|X|$, so $|\mathcal{C}|$ is even. We construct an instance of MPkRCO consisting of a singularly rooted mixed graph $G = (\mathbf{V}_X \cup \mathbf{V}_\mathcal{C} \cup r, \mathbf{A} \cup \mathbf{E})$ and some $T \subseteq V_X \cup V_\mathcal{C}$. First create a vertex r , let V_X consist of a set of vertices $\mathbf{S}_x = \{\mathbf{u}_x, \mathbf{v}_x, \mathbf{w}_x\}$ for every $x \in X$ and let $V_\mathcal{C}$ consist of one vertex \mathbf{a}_C for every $C \in \mathcal{C}$. Let A contain arcs $u_x v_x, v_x u_x, u_x w_x, k-1$ arcs from r to $w_x, k-1$ arcs from w_x to u_x and $k-1$ arcs from w_x to v_x for every $x \in X$ and $\max\{0, k-2\}$ arcs from r to a_C for every $C \in \mathcal{C}$. We next create E by adding an edge between u_x and a_C whenever $x \in C$, an edge between v_x and a_C whenever $\bar{x} \in C$ and one edge from r to a_C for all $C \in \mathcal{C}$. Finally, let $T = V_X$ if $k = 1$, $T = V_X \cup V_\mathcal{C}$ if $k \geq 3$ is odd and $T = \emptyset$ if k is even. Observe that, as $|\mathcal{C}|$ is even, for $k = 1$, we have $|A \cup E| = |A| + |E| = 3|X| + 4|X| + |\mathcal{C}| \equiv 3|X| = |T|$ and for $k \geq 2$, we have $|A \cup E| = |A| + |E| = 3k|X| + (k-2)|\mathcal{C}| + 4|X| + |\mathcal{C}| \equiv k|X| \equiv |T|$. In any case, we have $|A \cup E| \equiv |T|$.

We show that (G, T) is a positive instance of MPkRCO if and only if F is a positive instance of (3,B2)-SAT. An example can be found in Figure 3.1.6.

From assignment to orientation Suppose that F has a satisfying assignment. We create an orientation D of G in the following way. If a variable x is positive (negative) in the assignment, we orient the edges incident to $u_x(v_x)$ away from $u_x(v_x)$ and the edges incident to $v_x(u_x)$ toward $v_x(u_x)$. We orient the arcs between r and $V_\mathcal{C}$ so that $d_{A \cup E}^-(v_C)$ is odd if and only if $v_C \in T$. We show that this orientation satisfies the requirements.

Claim 3.1.8. *D is T -odd.*

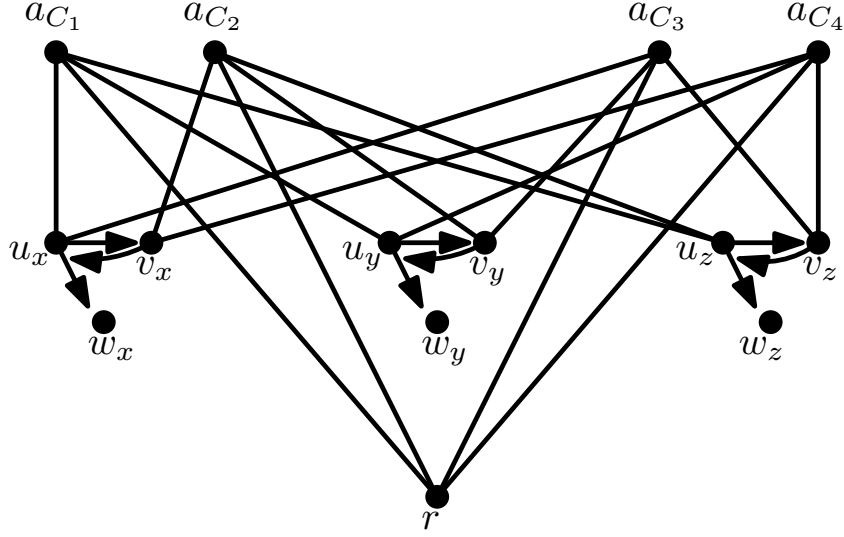


Figure 3.1.6: An example for G when the formula consists of the clauses $C_1 = \{x, y, z\}$, $C_2 = \{\bar{x}, \bar{y}, z\}$, $C_3 = \{x, \bar{y}, \bar{z}\}$ and $C_4 = \{\bar{x}, y, \bar{z}\}$ and $k = 1$.

Proof. First consider u_x for some $x \in X$. By construction, $d_{\bar{E}}^-(u_x)$ is even. It follows that $d_{A \cup \bar{E}}^-(u_x) \equiv d_A^-(u_x) = k$, which is odd if and only if $u_x \in T$. Similarly, $d_{A \cup \bar{E}}^-(v_x)$ is odd if and only if $v_x \in T$. Further, $d_{A \cup \bar{E}}^-(w_x) = d_A^-(w_x) = k$, which is odd if and only if $w_x \in T$. Further, we have that $d_{A \cup \bar{E}}^-(a_C)$ is odd if and only if $a_C \in T$ by construction. The statement now follows from the fact that $|T| \equiv |E|$ and Proposition 3.1.2. \square

The following observation crucially uses the fact that the assignment is satisfying.

Claim 3.1.9. *In D , there is at least one arc from V_X to a_C for every $C \in \mathcal{C}$.*

Proof. As the assignment is satisfying, there is at least one variable x that is set to the literal contained in C . By construction, the edge between S_x and a_C is oriented from S_x to a_C . \square

The following finishes the proof.

Claim 3.1.10. *D is k -root-connected from r .*

Proof. Suppose for the sake of a contradiction that there is a set $Y \subseteq V_C \cup V_X$ with $d_{A \cup \bar{E}}^-(Y) < k$. It can readily be seen that if $k \geq 2$ and Y contains one of u_x, v_x and w_x for some $x \in X$, then $S_x \subseteq Y$. If $k = 1$ and Y contains one of u_x, v_x and w_x for some $x \in X$, then Y contains u_x and v_x and we may suppose without loss of generality that Y also contains w_x , so $S_x \subseteq Y$. We may therefore consider the digraph $D' = (V_X' \cup V_C \cup r, A' \cup \bar{E}')$ that arises from contracting S_x to a vertex z_x for all $x \in X$ and obtain that this graph contains a set Y' with $d_{A' \cup \bar{E}'}^-(Y') < k$.

First consider the case that $k \geq 2$ and $Y' \subseteq V_C$. As there are no arcs between the vertices of V_C , we may suppose that $Y' = \{a_C\}$ for some $C \in \mathcal{C}$. By Claim 3.1.9 and construction, $d_{A' \cup \bar{E}'}^-(a_C) \geq k - 1$, so $d_{A' \cup \bar{E}'}^-(a_C) = k - 1$. This contradicts Claim 3.1.8. So suppose that Y' contains a vertex $z_x \in V_X$. By construction, $d_{A' \cup \bar{E}'}^-(r, z_x) = k - 1$ and there is an arc from some $a_C \in V_C$ to z_x . It follows that $a_C \in Y'$. By Claim 3.1.9, there is a variable $x' \in X - x$ such that there is an arc from $S_{x'}$ to a_C . It follows that $z_{x'} \in Y'$. Now $d_{A' \cup \bar{E}'}^-(Y') \geq d_{A'}(r, z_x) + d_{A'}(r, z_{x'}) = 2k - 2$, a contradiction as $k \geq 2$.

It remains to consider the case that $k = 1$. By construction, $d_{\bar{E}'}^-(z_x) = 2 = d_{\bar{E}'}^+(z_x)$ for all $x \in X$. Also, by Claims 3.1.8 and 3.1.9, we have $d_{\bar{E}'}^-(a_C) \geq 2 \geq d_{\bar{E}'}^+(a_C)$ for all

$C \in \mathcal{C}$. This yields $d_{A' \cup \vec{E}'}^-(Y') - d_{A' \cup \vec{E}'}^+(Y') = \sum_{v \in Y'} (d_{A' \cup \vec{E}'}^-(v) - d_{A' \cup \vec{E}'}^+(v)) \geq 0$, so as the underlying graph of D' is connected, Y' has at least one arc entering, a contradiction. \square

From orientation to assignment Suppose that G has a T -odd orientation D that is k -rooted connected from r . For every $x \in X$, as D is T -odd, either the edges entering $u_x(v_x)$ are both oriented toward $u_x(v_x)$ or both oriented away from $u_x(v_x)$. As $d_A^-(S_x) = k - 1$, at least one of these pairs of edges is oriented toward S_x . We now define an assignment by setting a variable x to TRUE if the edges incident to u_x are oriented away from u_x and FALSE otherwise. We will show that this assignment satisfies F . Consider some clause $C \in \mathcal{C}$. As D is T -odd and k -rooted connected from r , there is a directed edge from S_x to a_C for some $x \in X$. It follows that this variable has been assigned the value of the literal contained in C , so C is satisfied by the assignment.

Global connectivity This construction can easily be easily modified to obtain a hardness reduction for the problem when rooted connectivity is replaced by global connectivity. In order to see this, add an appropriate number of arcs from every vertex in $V_X \cup V_C$ to r . In particular, the problem of finding a strongly connected T -odd orientation of a mixed graph is NP-complete.

3.1.3.3 Open algorithmic problems

One approach to generalize PSCO is to replace the condition of the orientation being strongly connected by some higher connectivity condition. The following question can be asked for any integer $k \geq 1$.

Research Problem 3.1.7. *Let $G = (V, E)$ be a graph and $T \subseteq V$. What is the complexity of deciding whether G has a T -odd k -arc-connected orientation?*

Observe that for $k = 1$, the above problem is the same as PSCO. It is interesting to see that the proof of Lemma 3.1.9 does not generalize for $k \geq 2$, so it is not clear whether a reduction similar to Theorem 3.1.25 is available for any $k \geq 2$. This can be considered an indication that the problem is strictly harder for $k \geq 2$ than it is for $k = 1$. An even harder version of the problem can be obtained when k is considered part of the input rather than a fixed constant.

Research Problem 3.1.8. *Let $G = (V, E)$ be a graph, $T \subseteq V$ and $k \geq 1$ an integer. What is the complexity of deciding whether G has a T -odd k -arc-connected orientation?*

A different approach is to combine parity and well-balanced orientations.

Research Problem 3.1.9. *What is the complexity of deciding, given a graph $G = (V, E)$ and some $T \subseteq V$, whether G has a T -odd well-balanced orientation?*

While we expect this problem to be hard, the following can be established using a technique similar to the one applied in the proof of Theorem 3.1.26.

Theorem 3.1.29. *There is a polynomial time algorithm to decide, given a graph $G = (V, E)$ and some $T \subseteq V$, whether G has a T -odd best-balanced orientation.*

Finally, one can consider a problem in which the parity condition is relaxed to a subset of the vertices.

Research Problem 3.1.10. *Let $G = (V, E)$ be a graph and $T, U \subseteq V$ with $T \cap U = \emptyset$. What is the complexity of deciding whether G has a strongly connected orientation \vec{G} such that $d_{\vec{G}}^-(v)$ is odd for all $v \in T$ and $d_{\vec{G}}^-(v)$ is even for all $v \in U$?*

Clearly, when $T \cup U = V$, we are back to PSCO, so the above problem is more general.

3.1.4 Orientations of graphs of odd edge-connectivity

In this section, we prove some additional orientability properties that graphs of odd edge-connectivity have. This work is based on the results in [60].

Theorem 3.1.3 states that a graph has a k -arc-connected orientation if and only if it is $2k$ -edge-connected. When raising the edge-connectivity from an even number to an odd number by one, Theorem 3.1.3 does not yield any orientations with additional properties. We here introduce a graph invariant that is defined on graphs of odd edge-connectivity and describes an additional connectivity property of their orientations. We next describe the problem more formally. We say that an arc is *deletable* in a k -arc-connected orientation of a $(2k + 1)$ -edge-connected graph if its deletion leaves it k -arc-connected. We ask about the minimum number of orientations such that each edge of the original graph becomes a deletable arc in at least one of the orientations. Observe that the condition of the graph being $(2k + 1)$ -edge-connected cannot be replaced by $2k$ -edge-connectivity as the arcs associated to a $2k$ -edge-cut of G can never be deletable in a k -arc-connected orientation of G . Surprisingly, the number of necessary orientations is bounded by a constant depending only upon k . This is a consequence of a theorem of DeVos, Johnson and Seymour [23]. We focus on the case $k = 1$, meaning we want to find orientations of a 3-edge-connected graph such that for every edge of the graph, the deletion of the associated arc leaves a strongly connected graph in at least one of the orientations. In honor of András Frank who proposed this problem and had an immense impact on the development of the theory of graph orientations, we call the minimum number of necessary orientations for a graph G its *Frank number* $f(G)$. An example can be found in Figure 3.1.7.

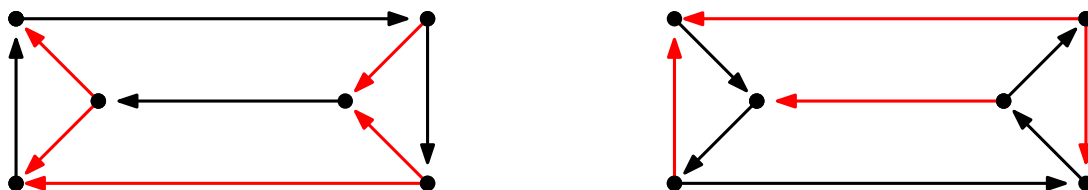


Figure 3.1.7: These two orientations of the so-called prism show that the Frank number of the prism is at most 2. In the two orientations of the prism, the deletable arcs are marked in red. Every edge becomes a deletable arc in one of the orientations.

Observe that the Frank number of any 4-edge-connected graph is 1 as it has a 2-arc-connected orientation by the theorem of Nash-Williams. On the other hand, any 3-edge-connected graph G containing a 3-edge-cut has Frank number at least 2. This follows directly from the fact that in any strongly connected orientation of G , there is one arc of the 3-edge-cut that is oriented differently than the other two arcs. This arc cannot be deletable in this orientation, so at least one more orientation is needed. It is an interesting question to find upper bounds for the Frank number of graphs. A first constant bound can easily be obtained by the following theorem of DeVos, Johnson and Seymour [23]:

Theorem 3.1.30. *Let $G = (V, E)$ be a 3-edge-connected graph. Then there is a partition $\{E_1, \dots, E_9\}$ of E such that $G - E_i$ is 2-edge-connected for all $i = 1, \dots, 9$.*

This implies the following:

Corollary 3.1.4. *Every 3-edge-connected graph G satisfies $f(G) \leq 9$.*

Indeed, by Robbins' Theorem, for all $i = 1, \dots, 9$, there is a strongly connected orientation of $G - E_i$. Giving an arbitrary orientation to the edges of E_i yields an orientation in which the arcs of \vec{E}_i are deletable.

The main contribution of this section is to further narrow down the values attained by the Frank number. We first show a better upper bound.

Theorem 3.1.31. *Every 3-edge-connected graph G satisfies $f(G) \leq 7$.*

In an attempt to improve on this, we also establish a relationship between our problem and a well-known conjecture about matchings in cubic graphs, the conjecture of Berge-Fulkerson mentioned in Section 3.1.4.1.

Theorem 3.1.32. *Every 3-edge-connected graph G satisfies $f(G) \leq 5$ unless the conjecture of Berge-Fulkerson fails.*

Further, we prove a stronger bound for two more restricted classes of 3-edge-connected graphs.

Theorem 3.1.33. *Every 3-edge-connected 3-edge-colorable graph G satisfies $f(G) \leq 3$.*

Theorem 3.1.34. *Every essentially 4-edge-connected graph G satisfies $f(G) \leq 3$.*

For the lower bound, we show that there are graphs whose Frank number is strictly bigger than 2, more precisely:

Theorem 3.1.35. *The Frank number of the Petersen graph is 3.*

Given a directed graph D , we call a set $F \subseteq A(D)$ *deletable* if $D - f$ is strongly connected for all $f \in F$. Given a graph G , we call a set $F \subseteq E(G)$ *deletable* if there exists an orientation \vec{G} of G such that \vec{F} is deletable in \vec{G} .

One of the main difficulties in improving the upper bound on the Frank number consists in finding a useful class of deletable sets. We consider the problem of testing algorithmically whether a set is deletable. More formally, we define the following problem:

DELETABILITY

Instance: A graph $G = (V, E)$ and a set $S \subseteq E$.

Question: Is there an orientation D of G such that $D - \vec{s}$ is strongly connected for all $s \in S$?

The following result shows that an efficient algorithm for DELETABILITY seems out of reach. This implies that a good characterization of deletable sets is hard to obtain.

Theorem 3.1.36. *DELETABILITY is NP-complete for cubic 3-edge-connected graphs.*

In Section 3.1.4.1, we present several classical results we will make use of and make some preparatory observations. Also, we introduce an auxiliary graph that will help to reduce the problems to cubic graphs later. In Section 3.1.4.2, we deal with the general case of 3-edge-connected graphs proving Theorems 3.1.31, 3.1.32 and 3.1.33. Section 3.1.4.3 is concerned with essentially 4-edge-connected graphs, in particular the proof of Theorem 3.1.34. In Section 3.1.4.4, we prove Theorem 3.1.35. Theorem 3.1.36 is proven in Section 3.1.4.5. Finally, in Section 3.1.4.6 we conclude our work and give directions for further research on this topic.

3.1.4.1 Preliminaries

We first mention the Conjecture of Berge and Fulkerson that Theorem 3.1.32 is based on. We then give two preliminary results that will be used for the main proofs in Sections 3.1.4.2 and 3.1.4.3. Finally, we give a construction that allows to reduce our problem to cubic graphs.

3.1.4.1.1 The conjecture of Berge and Fulkerson

We here mention mention an intensively studied conjecture which was proposed independently by Berge and Fulkerson [102]. For more information on this conjecture see [97].

Conjecture 3.1.2. *Every cubic 2-edge-connected graph has a set of six perfect matchings such that every edge is contained in exactly two of them.*

3.1.4.1.2 Preparatory results

We first give a result that describes the structure of 3-edge-cuts in a 3-edge-connected graph. It can be found as Theorem 7.1.2 in [41].

Proposition 3.1.3. *Let G be a 3-edge-connected graph and let $\delta(X)$ and $\delta(X')$ be 3-edge-cuts of G . Then $\delta(X)$ and $\delta(X')$ are not crossing, i.e. one of $X - X'$, $X' - X$, $X \cap X'$ or $V(G) - (X \cup X')$ is empty.*

The following characterization of deletable sets in digraphs is an immediate corollary of the definition of such sets.

Proposition 3.1.4. *Given a directed graph $D = (V, A)$, a set $F \subseteq A$ is deletable if and only if $\delta_D^-(X)$ contains either at least one arc of $A - F$ or at least two arcs for every nonempty, proper subset X of V .*

We next give a result that guarantees the existence of some deletable arcs in certain digraphs.

Lemma 3.1.10. *Let D be a strongly connected orientation of a 3-edge-connected graph G and C a circuit in D . Then C contains an arc a such that $D - a$ is strongly connected.*

Proof. Let $(\mathbf{G}, \mathbf{D}, \mathbf{C})$ be a counterexample that minimizes the number of vertices of D . Let e be an edge of G that is incident to a vertex of C and that does not belong to C . By the 3-edge-connectivity of G , e exists. Since D is strongly connected, \vec{e} belongs to a directed path P whose endvertices belong to C but whose internal vertices do not. By symmetry, we may suppose that the tail of \vec{e} is in $V(C)$. Then P can be extended by a possibly trivial directed subpath of C to form a circuit \mathbf{C}^* .

If $V(\mathbf{C}^*) = V$, let \mathbf{a} be the arc on C that has the same tail as \vec{e} . Clearly, we have $\mathbf{a} \in A(C) - A(\mathbf{C}^*)$. Further, as \mathbf{C}^* is strongly connected and \mathbf{C}^* is a subgraph of $D - \mathbf{a}$ on the same vertex set, we obtain that $D - \mathbf{a}$ is strongly connected.

We may hence suppose that $V(\mathbf{C}^*) \neq V$. Let $(\mathbf{G}', \mathbf{D}', \mathbf{C}')$ be obtained from (G, D, C) by contracting \mathbf{C}^* . Then, by Propositions 2.1.2(a) and 2.1.3(a), the assumptions of the lemma are satisfied for $(\mathbf{G}', \mathbf{D}', \mathbf{C}')$. By the minimality of (G, D, C) , \mathbf{C}' contains an arc \mathbf{a}' such that $D' - \mathbf{a}'$ is strongly connected. Let a be the arc of C in D that corresponds to \mathbf{a}' . Since $D' - \mathbf{a}'$ and \mathbf{C}^* are strongly connected, by Proposition 2.1.3(b), so is $D - a$. \square

3.1.4.1.3 Cubic extensions

We introduce for any graph $G = (V, E)$ of minimum degree at least 3 an auxiliary graph H_G that is cubic. For each $v \in V$ of degree at least 4, H_G contains a set \mathbf{S}_v of $d_G(v)$ vertices. For each $v \in V$ of degree 3, let $S_v = \{v\}$. Next, for each $v \in V$ of degree at least 4, we add a cycle \mathbf{C}_v whose vertex set is S_v . Finally, for each edge $uv \in E$, we add an edge between S_u and S_v to H_G . We do this in a way so that H_G becomes cubic. We call H_G a *cubic extension* of G . Note that H_G is not unique. This ambiguity has no consequences though.

Proposition 3.1.5. *Let $G = (V, E)$ be a graph of minimum degree at least 3 and H_G be a cubic extension of G .*

- (a) *If G is 3-edge-connected and $G - v$ is connected for all $v \in V$, then H_G is 3-edge-connected.*
- (b) *If G is essentially 4-edge-connected and $G - v$ is 2-edge-connected for all $v \in V$, then H_G is essentially 4-edge-connected.*

Proof. (a) Assume for a contradiction that $d_{H_G}(X) \leq 2$ for some nonempty, proper subset X of $V(H_G)$. Since G is 3-edge-connected, there is at least one $v \in V$ such that $S_v \cap X$ and $S_v - X$ are nonempty. It follows that $2 \leq d_{C_v}(X) \leq d_{H_G}(X) \leq 2$. This yields that for every $u \in V - v$ we have $S_u \subseteq X$ or $S_u \subseteq V(H_G) - X$ and for all $uw \in E$ with $v \notin \{u, w\}$ we have $S_u \cup S_w \subseteq X$ or $S_u \cup S_w \subseteq V(H_G) - X$. If there are vertices $u, w \in V$ such that $S_u \subseteq X$ and $S_w \subseteq V(H_G) - X$, it follows that $G - v$ is not connected, contradicting the assumption. Therefore, by symmetry we may assume that X is a nonempty, proper subset of S_v . We then have $d_{C_v}(X) \geq 2$ and there is at least one additional edge between X and $V(H_G) - S_v$, a contradiction to $d_{H_G}(X) \leq 2$.

(b) By (a), H_G is 3-edge-connected. For the sake of a contradiction, suppose that there is some nonsingular, proper subset X of $V(H_G)$ with $|V(H_G) - X| \geq 2$ such that $d_{H_G}(X) = 3$. If there are two vertices $u, v \in V$ such that $S_u \cap X$, $S_u - X$, $S_v \cap X$ and $S_v - X$ are nonempty, we have $2 + 2 \leq d_{C_u}(X) + d_{C_v}(X) \leq d_{H_G}(X) \leq 3$, a contradiction.

Now consider the case that there is exactly one $v \in V$ such that $S_v \cap X$ and $S_v - X$ are nonempty. We have that $d_{H_G}(X) - d_{C_v}(X) \leq 1$. It follows that in H_G there is at most one edge between $X - S_v$ and $V(H_G) - X - S_v$. If $X - S_v$ and $V(H_G) - X - S_v$ are nonempty, then $G - v$ is not 2-edge-connected, a contradiction to the assumption. By symmetry, we may therefore assume that $X \subseteq S_v$. We have that $d_{C_v}(X) = 2$ and there are $|X|$ edges between X and $V(H_G) - S_v$. It follows that $|X| = 1$, which is a contradiction.

Finally assume that $S_v \subseteq X$ or $S_v \cap X = \emptyset$ for all $v \in V$. Let $X' = \{v \in V : S_v \subseteq X\}$. As G is essentially 4-edge-connected, we may assume by symmetry that $X' = \{v\}$ for some vertex v of degree 3. This yields that $|X| = |S_v| = 1$, which is a contradiction. \square

3.1.4.2 3-edge-connected graphs

This section is dedicated to proving Theorems 3.1.31, 3.1.32 and 3.1.33. In the first part, we show that a certain class of edge sets is deletable. After, we show how to cover the edge sets of cubic 3-edge-connected graphs with such sets. Next, we use this to conclude cubic versions of Theorems 3.1.31 and 3.1.32 and to prove Theorem 3.1.33. Finally, we extend this to obtain the general versions of Theorems 3.1.31 and 3.1.32.

3.1.4.2.1 A class of deletable edge sets

Given a packing \mathcal{C} of cycles in a 3-edge-connected graph G , the *special set* of \mathcal{C} is defined to be the set of edges in $E(G) - E(\mathcal{C})$ that belong to no 3-edge-cut of G/\mathcal{C} . An example can be found in Figure 3.1.8.

Lemma 3.1.11. *Let M be the special set of a cycle packing \mathcal{C} of a 3-edge-connected graph G . Then M is deletable.*

Proof. Let $G' = G/\mathcal{C}$ and let M' be the set of edges in M that exist in G' . Since G is 3-edge-connected, so is G' by Proposition 2.1.2(a). Consider a well-balanced orientation D' of G' which exists by Theorem 3.1.6. Then D' is strongly connected. Let D be the orientation of G obtained from D' by orienting all cycles of \mathcal{C} as a circuit and giving an arbitrary orientation to all remaining edges that do not exist in G' .

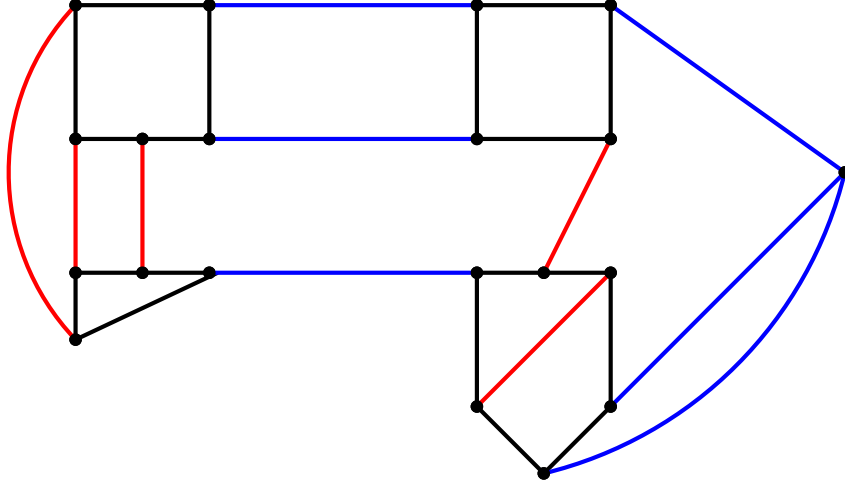


Figure 3.1.8: An example for a special set. The edges marked in black form a cycle packing whose special set is marked in red.

By Proposition 2.1.3 (b), D is strongly connected. We have to show that $D - \vec{f}$ is strongly connected for all $f \in M$. By construction, that is the case for all $f \in M - M'$, so, by Proposition 2.1.3(b), it is enough to show that $D' - \vec{f}$ is strongly connected for all $f \in M'$. Let $\vec{f} = uv$ for some $f \in M$ and suppose that there exists some nonempty, proper subset X of $V(D')$ with $|\delta_{D' - \vec{f}}^+(X)| = 0$. Obviously $u \in X$ and $v \in V(D') - X$. Since G' is 3-edge-connected and f belongs to no 3-edge-cut in G' , we obtain $\lambda_{G'}(u, v) \geq 4$. As D' is well-balanced, it follows that $0 = |\delta_{D' - \vec{f}}^+(X)| = |\delta_{D'}^+(X)| - 1 \geq \lfloor \frac{\lambda_{G'}(u, v)}{2} \rfloor - 1 \geq 2 - 1 = 1$, a contradiction. \square

3.1.4.2.2 Covering cubic graphs with special sets

In the following we show that any cubic 3-edge-connected graph can be covered by 7 special sets. For technical reasons, we will need the following slight strengthening.

Lemma 3.1.12. *For every cubic 3-edge-connected graph, there exist 7 cycle packings satisfying the following conditions:*

- (a) *Every edge is in the special set of at least one cycle packing.*
- (b) *Every edge is in exactly 4 of the cycle packings.*

Proof. For the sake of a contradiction, let $G = (V, E)$ be a counterexample to the lemma that minimizes $|V|$.

Claim 3.1.11. *G is essentially 4-edge-connected.*

Proof. For the sake of a contradiction, let $\{\mathbf{S}_1, \mathbf{S}_2\}$ be a partition of $V(G)$ such that $|S_i| \geq 2$ and a 3-edge-cut $\mathbf{F} := \{e_1, e_2, e_3\}$ exists between S_1 and S_2 . Construct the graphs G_i from G by contracting S_{3-i} to v_i . As G_i is cubic, 3-edge-connected by Proposition 2.1.2(a) and smaller than G , there exists a set of cycle packings $\mathbb{C}^i = \{\mathcal{C}_1^i, \dots, \mathcal{C}_7^i\}$ of G_i satisfying (a) and (b).

Observe that since G_i is cubic, (b) implies that for $j \in \{1, 2, 3\}$, there are exactly two cycle packings in \mathbb{C}^i that contain $\{e_1, e_2, e_3\} - \{e_j\}$. It follows that v_i is in exactly 6 cycle packings of \mathbb{C}^i . By relabeling if needed, we may assume that \mathcal{C}_1^i is the cycle packing that does not contain v_i and $\{e_1, e_2, e_3\} - \{e_j\}$ is contained in \mathcal{C}_{2j}^i and \mathcal{C}_{2j+1}^i . We may also assume, by (a), that e_j is in the special set of \mathcal{C}_{2j}^i .

We construct $\mathbb{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_7\}$ so that $E(\mathcal{C}_k) = E(\mathcal{C}_k^1) \cup E(\mathcal{C}_k^2)$ for $k = 1, \dots, 7$. Observe that this is a set of seven cycle packings. We finish the proof by showing that \mathbb{C} satisfies (a) and (b).

First observe that (b) follows directly from the construction and the fact that an edge is in \mathcal{C}_k if and only if it is in \mathcal{C}_k^1 or \mathcal{C}_k^2 . For (a), let first e be an edge in $G[S_i]$. By (a), there exists a $k \in \{1, \dots, 7\}$ such that e is in the special set of \mathcal{C}_k^i . First observe that e is in $E(G) - E(\mathcal{C}_k)$. If e is in a 3-edge-cut F' of G/\mathcal{C}_k , since F' is not a 3-edge-cut of G_i/\mathcal{C}_k^i , F' contains an edge of $G[S_{3-i}]$. This yields that F and F' are crossing 3-edge-cuts of G , a contradiction to Proposition 3.1.3.

Now consider the edge e_j for some $j \in \{1, 2, 3\}$. As $e_j \in E(G_i) - E(\mathcal{C}_{2j}^i)$, we have $e_j \in E(G) - E(\mathcal{C}_{2j})$. Again, assume that e_j is in a 3-edge-cut F' of G/\mathcal{C}_{2j} . As F' is not a 3-edge-cut in G_1/\mathcal{C}_{2j}^1 and G_2/\mathcal{C}_{2j}^2 , we obtain that F' and F are crossing in G contradicting Proposition 3.1.3. This finishes the proof of the claim. \square

By Theorem 2.1.7, G contains a perfect matching M . Since G is cubic, the connected components of $G - M$ form a cycle packing \mathcal{C}_1 . Now consider the graph G' which is obtained from G by contracting every cycle of $C \in \mathcal{C}_1$ into a vertex v_C and adding a loop at v_C for every edge in M that has both ends in $V(C)$. Further, let T be its set of odd-degree vertices.

Claim 3.1.12. *The edge set of G' can be partitioned into three T -joins F_1 , F_2 and F_3 .*

Proof. As G is essentially 4-edge-connected by Claim 3.1.11, G' is essentially 4-edge-connected by Proposition 2.1.2(b). As G is essentially 4-edge-connected, it follows that $d_{G'}(v_C) \geq d_G(C) \geq 4$ for every $C \in \mathcal{C}_1$, so G' is 4-edge-connected. By Theorem 2.1.6, there exist two edge-disjoint spanning trees F'_1, F'_2 of G' . By Proposition 2.1.8, each of them contains a T -join $F_i, i = 1, 2$. As $F_1 \cup F_2$ is Eulerian, $F_3 = E(G') - F_1 - F_2$ is also a T -join. \square

Claim 3.1.13. *For $i = 1, 2, 3$, there exist V -joins S_{2i} and S_{2i+1} of G such that $S_{2i} \cap S_{2i+1} = F_i$ and $S_{2i} \cup S_{2i+1} = (E - M) \cup F_i$.*

Proof. For $i = 1, 2, 3$, let T_i be the set of vertices in V not incident to an edge in F_i . Let $C \in \mathcal{C}_1$. Observe that, as G is cubic and $F_i \subseteq M$ is a matching in G , we obtain $|V(C)| \equiv d_G(V(C)) = d_{G'}(v_C)$ and $|V(C) \cap V(F_i)| \equiv d_{F_i}(V(C)) = d_{F_i}(v_C)$. As F_i is a T -join in G' , this yields $|T_i \cap V(C)| = |V(C)| - |V(C) \cap V(F_i)| \equiv d_{G'}(v_C) - d_{F_i}(v_C) \equiv 0$, so $|T_i \cap V(C)|$ is even. Hence, by Proposition 2.1.8, we obtain that $G - M$ contains a T_i -join N_i . Let $S_{2i} := F_i \cup N_i$ and $S_{2i+1} := F_i \cup (E - M - N_i)$. By construction, we have that S_{2i} and S_{2i+1} are V -joins in G such that $S_{2i} \cap S_{2i+1} = F_i$ and $S_{2i} \cup S_{2i+1} = (E - M) \cup F_i$. \square

For $j = 2, \dots, 7$, we define \mathcal{C}_j to be the set of nonsingular connected components of $G - S_j$. Observe that all of them are cycles as S_j is a V -join and G is cubic.

Claim 3.1.14. *$\mathcal{C}_1, \dots, \mathcal{C}_7$ satisfy (a) and (b).*

Proof. (a) For $e \in M$, since G is essentially 4-edge-connected, e is in the special set of \mathcal{C}_1 .

For $e \in E - M$, let f and g be the two edges of M adjacent to e . Since F_1, F_2 and F_3 are disjoint, there is an F_i that contains neither f nor g . Then, since G is cubic and by Claim 3.1.13, one of the V -joins S_{2i} and S_{2i+1} , say S_j , contains e but none of the edges adjacent to e . It follows that both endvertices of e in G are in cycles of \mathcal{C}_j . As G is essentially 4-edge-connected, it follows that both endvertices of e in G/\mathcal{C}_j are of degree at least 4. As G is essentially 4-connected, so is G/\mathcal{C}_j by Proposition 2.1.2(b). This yields that e is in no 3-edge-cut of G/\mathcal{C}_j and so e is in the special set of \mathcal{C}_j .

(b) For $e \in M$, by Claim 3.1.12, e is in exactly one F_i , say F_1 . Then, by Claim 3.1.13, e is in $\mathcal{C}_4, \dots, \mathcal{C}_7$ and not in $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$.

For $e \in E - M$, e is in \mathcal{C}_1 and, by Claim 3.1.13, in exactly one of \mathcal{C}_{2i} and \mathcal{C}_{2i+1} for $i = 1, 2, 3$. \square

Claim 3.1.14 finishes the proof of Lemma 3.1.12. \square

3.1.4.2.3 Cubic case

We first show how to conclude a cubic version of Theorem 3.1.31.

Theorem 3.1.37. *Let G be a cubic 3-edge-connected graph. Then $f(G) \leq 7$.*

Proof. Lemma 3.1.12 yields that $E(G)$ can be covered by 7 special sets $\mathbf{S}_1, \dots, \mathbf{S}_7$. By Lemma 3.1.11, there exist orientations $\mathbf{D}_1, \dots, \mathbf{D}_7$ of G such that S_i is deletable in D_i for $i = 1, \dots, 7$. It follows that the Frank number of G is at most 7. \square

Next, we use Lemma 3.1.11 to show that perfect matchings with a certain additional property are deletable. As corollaries, we obtain Theorem 3.1.33 and a cubic version of Theorem 3.1.32.

Lemma 3.1.13. *Let M be a perfect matching of a cubic 3-edge-connected graph G intersecting every 3-edge-cut of G in exactly one edge. Then M is deletable.*

Proof. As G is cubic and M is a perfect matching of G , the connected components of $G - M$ form a packing \mathcal{C} of cycles. We show that G/\mathcal{C} is 4-edge-connected. By Proposition 2.1.3(a) and since G is 3-edge-connected, so is G/\mathcal{C} . A 3-edge-cut of G/\mathcal{C} would provide a 3-edge-cut of G intersecting M in 3 edges contradicting the assumption. It follows that M is the special set of \mathcal{C} and therefore deletable by Lemma 3.1.11. \square

We first show how to conclude Theorem 3.1.33 from Lemma 3.1.13.

Proof. (of Theorem 3.1.33) Let \mathbf{G} be a 3-edge-colorable 3-edge-connected graph. Then G is cubic and has 3 disjoint perfect matchings $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$ covering the edge set of G . Let $\delta(X)$ be a 3-edge-cut of G . Since G is cubic and $d(X) = 3$, we obtain that $|X|$ is odd. Then, since M_i is a perfect matching, we obtain that $\delta(\mathbf{X})$ intersects each M_i . As $d(X) = 3$ and the matchings are disjoint, we obtain that $\delta(X)$ intersects each of M_1, M_2, M_3 exactly once. It follows by Lemma 3.1.13 that each of M_1, M_2, M_3 is deletable, so $f(G) \leq 3$. \square

Next, we prove in a similar way the following cubic version of Theorem 3.1.32.

Theorem 3.1.38. *Let G be a cubic 3-edge-connected graph that satisfies Conjecture 3.1.2. Then $f(G) \leq 5$.*

Proof. By assumption, there exist 6 perfect matchings $\mathbf{M}_1, \dots, \mathbf{M}_6$ of G covering each edge of G exactly twice.

Let $\delta(\mathbf{X})$ be a 3-edge-cut of G . Since G is cubic and $d(X) = 3$, we obtain that $|X|$ is odd. Then, since M_i is a perfect matching, $\delta(X)$ intersects each M_i . Since each of the 3 edges of $\delta(X)$ belongs to exactly 2 M_i 's, $\delta(X)$ intersects each of M_1, \dots, M_6 exactly once. It follows by Lemma 3.1.13 that each of M_1, \dots, M_6 is deletable. As every edge of G is covered by at least one of M_1, \dots, M_5 , it follows that $f(G) \leq 5$. \square

3.1.4.2.4 Non-cubic case

We first show how to prove the general case of Theorem 3.1.31.

Proof. (of Theorem 3.1.31) Let G be a counterexample minimizing $|V(G)|$.

Claim 3.1.15. G is 2-vertex-connected.

Proof. For the sake of a contradiction, assume that G has a cut vertex v . So G has two nonsingular subgraphs G_1 and G_2 such that $G_1 = G/G_2$ and $G_2 = G/G_1$. As G is 3-edge-connected, so is G_i by Proposition 2.1.2(a). Since G_i is smaller than G , G_i has Frank number at most 7. So there exist 7 orientations D_j^i of G_i such that for each edge e of G_i , one of $D_j^i - \vec{e}$ is strongly connected. We can now construct the 7 orientations D_j of G by giving each edge in G_i its orientation in D_j^i also in D_j . Now consider an edge e of G_i and let $D_j^i - \vec{e}$ be strongly connected. Since $D_j^i - \vec{e} = (D_j - \vec{e})/D_j^{3-i}$ and D_j^{3-i} are strongly connected, Proposition 2.1.3(b) implies that so is $D_j - \vec{e}$. It follows that G has Frank number at most 7, a contradiction. \square

Let H_G be a cubic extension of G as defined in Section 3.1.4.1.3. By Claim 3.1.15 and Proposition 3.1.5(a), H_G is 3-edge-connected. Then, by Theorem 3.1.37, the Frank number of H_G is at most 7, that is there exist 7 orientations D'_i of H_G such that for each edge e of H_G , one of $D'_i - \vec{e}$ is strongly connected. Let D_i be the orientation of G obtained from D'_i by contracting the subgraphs C_v for all $v \in V(G)$. For any $e \in E(G) \subseteq E(H_G)$, one of $D'_i - \vec{e}$ is strongly connected, therefore, by Proposition 2.1.3(a), so is $D_i - \vec{e}$. It follows that the Frank number of G is at most 7, a contradiction. \square

The same reduction and Theorem 3.1.38 show Theorem 3.1.32.

3.1.4.3 Essentially 4-edge-connected graphs

This section is dedicated to proving Theorem 3.1.34. Again, first we prove the result for cubic graphs and then we show how it implies the non-cubic case.

3.1.4.3.1 Cubic case

In the case of essentially 4-edge-connected graphs, we can show that every matching is deletable. We prove the following slightly stronger statement.

Lemma 3.1.14. *Let G be an essentially 4-edge-connected graph, M a matching of G and \mathcal{C} a cycle packing of $G - M$. Then there exists an orientation of G in which \vec{M} is deletable and each cycle of \mathcal{C} is oriented as a circuit.*

Proof. Let \mathcal{F} be the set of maximal 2-edge-connected subgraphs of $G - M$. Let $G' = (V', E')$ be the graph obtained from G by contracting each graph of \mathcal{F} and let $M' = M \cap E'$. Note that $G' - M'$ is a forest. Since G is essentially 4-edge-connected, by Proposition 2.1.2(b), so is G' and every vertex of degree 3 in G' is an original vertex of G . Then, since M is a matching of G , every vertex v of degree 3 in G' is incident to at least 2 edges e_v^1, e_v^2 in $E' - M'$.

By Theorem 3.1.7 and Lemma 3.1.1, there exists an orientation-admissible pairing P of G' . As $G' \cup P$ is Eulerian, Proposition 2.1.12 yields that $G' \cup P$ has an Eulerian orientation $\vec{G}' \cup \vec{P}$ such that for each vertex v of degree 3 in G' , one of \vec{e}_v^1, \vec{e}_v^2 enters v

and the other one leaves v . By the definition of orientation-admissible pairings, \vec{G}' is a well-balanced orientation of G' .

For all $F \in \mathcal{F}$, by Proposition 2.1.2(a), Theorem 3.1.2 and Proposition 2.1.3(b), there exists a strongly connected orientation \vec{F} of F such that each cycle of \mathcal{C} contained in F is oriented as a circuit.

Let \vec{G} be the orientation of G obtained by combining \vec{G}' and \vec{F} for all $F \in \mathcal{F}$ and giving an arbitrary orientation to the arcs in $M - M'$. Proposition 2.1.3(b) yields that \vec{G} is strongly connected. Since each cycle C of \mathcal{C} belongs to some $F \in \mathcal{F}$, C is oriented as a circuit in \vec{G} .

We will finish the proof by showing that $\vec{G} - \vec{e}$ is strongly connected for all $e \in M$. Since \vec{F} is strongly connected for all $F \in \mathcal{F}$ and $\bigcup_{F \in \mathcal{F}} E(F)$ contains no edge in M , it suffices to prove, by Proposition 2.1.3(b), that $\vec{G}' - \vec{e}$ is strongly connected for all $e \in M'$. Let X be a subset of V' . By Proposition 3.1.4, it is enough to prove that either at least two arcs or at least one arc of $\vec{E}' - \vec{M}'$ leave X .

If there are $x \in X$ and $y \in V' - X$ of degree at least 4, then, since G' is essentially 4-edge-connected, there is no 3-edge-cut separating x and y in G' and therefore, as \vec{G}' is well-balanced, there are 2 arcs leaving X , so we are done.

Hence, by considering $V' - X$ and \vec{G}' if necessary, we may assume without loss of generality that X only contains vertices of degree 3 and there is no arc of \vec{E}' leaving X . By construction, every vertex v of X has at least one arc \vec{e}_v^1 or \vec{e}_v^2 of \vec{E}' leaving v . As there is no arc of \vec{E}' leaving X , we obtain that $\vec{G}'[X]$ contains a circuit \vec{C} of arcs in \vec{E}' by Proposition 2.1.6. The corresponding cycle C provides a contradiction since $G' - M$ is a forest. \square

We are now ready to prove a cubic version of Theorem 3.1.34.

Theorem 3.1.39. *Let G be a cubic essentially 4-edge-connected graph. Then $f(G) \leq 3$.*

Proof. Since G is cubic and 2-edge-connected, by Theorem 2.1.7, G has a perfect matching M_1 and the connected components of $G - M_1$ form a packing \mathcal{C} of cycles. By Lemma 3.2.16, there exists an orientation D_1 of G such that each cycle of \mathcal{C} is oriented as a circuit and M_1 is deletable in D_1 . By Lemma 3.1.10, each $C_i \in \mathcal{C}$ contains a deletable arc \vec{e}_i in D_1 . Note that the connected components of $G - M_1 - \cup\{e_i : C_i \in \mathcal{C}\}$ form a packing of paths which is the union of two matchings M_2 and M_3 . By Lemma 3.2.16, there exist orientations D_2 and D_3 of G such that M_2 is deletable in D_2 and M_3 is deletable in D_3 . Since $E(G) = M_1 \cup M_2 \cup M_3 \cup \{e_i : C_i \in \mathcal{C}\}$, Theorem 3.1.39 follows. \square

3.1.4.3.2 Non-cubic case

We now generalize the results of the previous part to arbitrary essentially 4-edge-connected graphs.

Proof. (of Theorem 3.1.34). Let $G = (V, E)$ be a counterexample minimizing $|V|$.

Claim 3.1.16. *$G - v$ is 2-edge-connected for all $v \in V$.*

Proof. For the sake of a contradiction, suppose that $G - v$ is not 2-edge-connected for some $v \in V$. If $G - v$ is disconnected, we obtain a contradiction using the same argument as in the proof of Claim 3.1.15. We therefore have a partition $\{\mathcal{S}_1, \mathcal{S}_2\}$ of $V - \{v\}$ such that S_1 and S_2 are only connected by a single edge e_0 in $G - v$. Let us denote the endvertices of e_0 by $\mathbf{u}_i \in S_i$. Consider the graph G_i that arises from G by contracting $A_{S-i} \cup \{v\}$ into a vertex \mathbf{v}_i . Note that $E(G_1) \cap E(G_2) = \{e_0\}$. Since G is essentially 4-edge-connected, so

is G_i by Proposition 2.1.2 (b). Moreover, G_i is smaller than G . It follows that there exist 3 orientations \mathbf{D}_j^i of G_i such that one of $D_j^i - \vec{e}$ is strongly connected for all $e \in E(G_i)$. We may suppose that $D_1^1 - \vec{e}_0$ and $D_1^2 - \vec{e}_0$ are strongly connected. Reversing the arcs in D_j^2 if needed, we may assume that e_0 has the same orientation in D_j^1 and D_j^2 . We can construct the 3 orientations \mathbf{D}_j of G by merging D_j^1 and D_j^2 . We will finish the proof by showing that for all $e \in E$, there exists a j such that $D_j - \vec{e}$ is strongly connected. Let $e \in E$ and $j \in \{1, 2, 3\}$ such that both $D_j^1 - \vec{e}$ and $D_j^2 - \vec{e}$ are strongly connected. Observe that if $e \neq e_0$, then either $D_j^1 - \vec{e} = D_j^1$ or $D_j^2 - \vec{e} = D_j^2$. Assume that there is a nonempty, proper subset X of V that has no arc leaving in $D_j - \vec{e}$. Without loss of generality, we may assume that $v \in X$. As $(X \cap S_i) \cup \{v_i\}$ has an arc leaving in $D_j^i - \vec{e}$, we obtain that $e = e_0$ and e_0 must be directed away from v_i in D_j^i for $i = 1, 2$. This is a contradiction as D_j^1 and D_j^2 were chosen to both have the same orientation of e_0 . \square

Let \mathbf{H}_G be a cubic extension of G as defined in Section 3.1.4.1.3. By Claim 3.1.16 and Proposition 3.1.5(b), H_G is a cubic essentially 4-edge-connected graph. Then, by Theorem 3.1.39, the Frank number of H_G is at most 3. There exist therefore 3 orientations \mathbf{D}'_j of H_G such that for each edge $e \in E(H_G)$, there is some $j \in \{1, 2, 3\}$ such that $D'_j - \vec{e}$ is strongly connected. Consider now the 3 orientations \mathbf{D}_j of G which arise from D'_j by contracting the subgraphs C_v for all $v \in V$. By Proposition 2.1.3(a), if $D'_j - \vec{e}$ is strongly connected for an edge $e \in E$, so is $D_j - \vec{e}$. It follows that the Frank number of G is at most 3, a contradiction. \square

3.1.4.4 The Petersen graph

In this section, we show that there are graphs of Frank number higher than two, more precisely we prove Theorem 3.1.35. While this result can also be established computationally, we prefer to give a proof by hand.

Proof. (of Theorem 3.1.35) Let $\mathbf{G} = (V, E)$ be the Petersen graph, see Figure 2.1.6. We frequently make use of the symmetry properties of G . By Theorem 3.1.34 and since G is essentially 4-edge-connected, but not 4-edge-connected, it suffices to prove that its Frank number is different from 2. Suppose that G has Frank number 2 and let $\mathbf{D}_1 = (V, A_1)$ and $\mathbf{D}_2 = (V, A_2)$ be two orientations of G such that

$$D_1 - \vec{e} \text{ or } D_2 - \vec{e} \text{ is strongly connected for each edge } e \text{ of } G. \quad (*)$$

We say that an arc of D_1 is *stable* if the same arc exists in D_2 , otherwise it is *changing*. Let \mathbf{S} and \mathbf{C} be the set of stable and changing arcs, respectively. Note that D_1 and D_2 also satisfy (*) and stable and changing arcs are exchanged. Hence, whatever is proved for stable arcs is also true for changing arcs.

We first show that \mathbf{S} and \mathbf{C} induce a 2-edge-coloring of G with certain properties and then that no such 2-edge-coloring exists. Observe that none of the considered colorings are required to be proper. For a 2-edge-coloring R, B of G , we define an auxiliary graph $\mathbf{H}^{R,B} := (V, F)$ where $uv \in F$ if there exists a 3-path $tuvw$ in $G(R)$ or in $G(B)$ or there exists a (u, v) -path that is a connected component of $G(R)$ or of $G(B)$.

Lemma 3.1.15. *G has a 2-edge-coloring R, B such that*

$$\text{no monochromatic 3-star exists,} \quad (3.1.1)$$

$$\mathbf{H}^{R,B} \text{ is bipartite.} \quad (3.1.2)$$

Proof. We show that the 2-edge-coloring induced by S and C satisfies (3.1.1) and (3.1.2). To show (3.1.1) we need the following claim.

Claim 3.1.17. *Each vertex is incident to at least one stable arc.*

Proof. Suppose that a vertex v is incident only to changing arcs. Since G is cubic and D_1 is strongly connected, either the in-degree or the out-degree of v is 1, say \vec{e} is the only arc entering v . Then \bar{e} is the only arc leaving v in D_2 . Then, $D_1 - \vec{e}$ and $D_2 - \bar{e}$ are not strongly connected, which is a contradiction. \square

To show (3.1.2) we need the following claims.

Claim 3.1.18. *The weakly connected components of $D_1(S)$ are directed paths or circuits.*

Proof. By Claim 3.1.17 applied for stable arcs and then for changing arcs, the connected components of $D_1(S)$ are paths or cycles. If two stable arcs are incident to a vertex v , then one of them enters and the other one leaves v . Otherwise, let e be the third arc incident to v . Then, $D_1 - \vec{e}$ and $D_2 - \bar{e}$ are not strongly connected, which is a contradiction. Now the claim follows. \square

Claim 3.1.19. *Let P be a weakly connected component of $D_1(S)$ that is a directed (u, v) -path. Then the in-degrees of u and v in D_1 are of different parity.*

Proof. Since G is cubic and u and v are incident to exactly one stable arc in D_1 , u and v are incident to exactly two changing arcs in D_1 . Then, by Claim 3.1.18 applied for $D_1(C)$, exactly one changing arc enters both u and v in D_1 . Since P is a directed path between u and v , the claim follows. \square

Claim 3.1.20. *Let $tuvw$ be a 3-path in $D_1(S)$. Then the in-degrees of u and v are of different parity in D_1 .*

Proof. By Claim 3.1.18, exactly one stable arc enters both u and v in D_1 . By Claim 5.2.4, the two other arcs incident to u and v are changing. If both are entering or leaving then $D_1 - uv$ and $D_2 - uv$ are not strongly connected, which is a contradiction. Now the claim follows. \square

Claim 3.1.21. *$H^{S,C}$ is a bipartite graph.*

Proof. Since G is cubic and D_1 and D_2 are strongly connected, each vertex is of in-degree 1 or 2. By Claims 3.1.19 and 3.1.20, each edge of $H^{S,C}$ is between a vertex of in-degree 1 and a vertex of in-degree 2, so $H^{S,C}$ is bipartite. \square

By Claim 5.2.4 applied for $R := S$ and $B := C$ and by Claim 3.1.21, Lemma 3.1.15 follows. \square

We show that G does not admit any 2-edge-coloring satisfying (3.1.1) and (3.1.2) and obtain a contradiction to Lemma 3.1.15.

The following result yields a strong property such a coloring would have to satisfy.

Lemma 3.1.16. *Let R, B be a 2-edge-coloring satisfying (3.1.1) and (3.1.2). Then G has a 5-cycle that contains a monochromatic 4-path whose endvertices are incident to 2 edges of the other color.*

Proof. We first show two weaker statements which are useful in the proof later on.

Claim 3.1.22. *G has a monochromatic 3-path.*

Proof. Suppose not. Since G is cubic, there are two adjacent edges of the same color, without loss of generality $ab, ae \in R$. Then, by the assumption for $deab, eabc, eabi$ and $heab$, we obtain that $de, bc, bi, eh \in B$. Thus, by the assumption for $cbih, cbij$ and $dehg$, we obtain that $jihg$ forms a monochromatic 3-path, contradicting the assumption. See Figure 3.1.9a. \square

This result is helpful in proving a strengthening of itself.

Claim 3.1.23. G has a 5-cycle that contains a monochromatic 4-path.

Proof. Suppose not. By Claim 3.1.22, without loss of generality $bc, cd, de \in B$. Then, by the assumption for $abcde$, we obtain that $ab, ae \in R$. By (3.1.1) for a, c and d , we obtain that $af \in B$ and $cg, dj \in R$. By (3.1.1) for f , one of fg and fj is in R . By symmetry, without loss of generality $fg \in R$. Then, by (3.1.1) for $g, gh \in B$. So, by the assumption for $cdehg$, we obtain that $eh \in R$. Then, by the assumption for $abihe$, we obtain that $hi, bi \in B$. Thus $cbihg$ forms a monochromatic 4-path in the 5-cycle $cbihg$, that contradicts the assumption. See Figure 3.1.9b. \square

By Claim 3.1.23, without loss of generality $ab, bc, de, ea \in R$. Then, by (3.1.2) for $abcde$, we obtain that $cd \in B$. By (3.1.1) for a, b and e , we obtain that $af, bi, eh \in B$. If $hi \in B$, then, by the 3-paths of $deabc$ and by $ehib$, we obtain that $H^{R,B}$ contains the 3-cycle abe that contradicts (3.1.2). See Figure 3.1.9c. Hence, $hi \in R$. If $cg, dj \in R$, then, by the 3-paths of $jdeabcg$ and by cd , we obtain that $H^{R,B}$ contains the 5-cycle $abcde$ that contradicts (3.1.2). See Figure 3.1.9d. Hence, by symmetry, we may suppose that $dj \in B$.

Now suppose for the sake of a contradiction that G does not contain a 5-cycle that contains a monochromatic 4-path whose endvertices are incident to 2 edges of the other color. If $cg \in B$, then $abcde$ contradicts the assumption. See Figure 3.1.9e. Hence $cg \in R$. If $ij \in B$, then $bcdji$ contradicts the assumption. See Figure 3.1.9f. Hence $ij \in R$. If $gh \in R$, then, by the 3-paths of $abcghij$ and by bi , we obtain that $H^{R,B}$ contains the 5-cycle $bcghi$ that contradicts (3.1.2). See Figure 3.1.9g. Hence $gh \in B$. If $fg \in B$, then $afghe$ contradicts the assumption. See Figure 3.1.9h. Hence $fg \in R$. Then, by the 3-paths of $deabcfg$ and by ehg , we obtain that $H^{R,B}$ contains the 5-cycle $abcge$ that contradicts (3.1.2). See Figure 3.1.9i. This finishes the proof of Lemma 3.1.16. \square

Lemma 3.1.16 yields that G has a 5-cycle, without loss of generality $abcde$, that contains a monochromatic 4-path whose endvertices are incident to 2 edges of the other color. By similar arguments as before, we obtain the partial coloring of Figure 3.1.9e. By (3.1.1) for f , one of fg and fj is in R . By symmetry, without loss of generality $fj \in R$.

Suppose that $fg \in B$. Then, by (3.1.1) for $g, gh \in R$. If $ij \in R$, then, by the 3-paths of $deabc$ and for $ghij$, and by eh and ib , $H^{R,B}$ contains the 5-cycle $eabih$ contradicting (3.1.2). See Figure 3.1.9j. If $ij \in B$, then, by the 3-paths of $afgcdji$ and by fj , we obtain that $H^{R,B}$ contains the 5-cycle $fgcdj$ contradicting (3.1.2). See Figure 3.1.9k.

Hence $fg \in R$. Then, by (3.1.2) for $fghij$, one of hg and ij is in B . By symmetry, we may suppose that $ij \in B$. If $hg \in R$, then, by the 3-paths of $jfghi$ and for $deab$, and by eh and af , we obtain that $H^{R,B}$ contains the 5-cycle $fghea$ contradicting (3.1.2). See Figure 3.1.9l. If $hg \in B$, then, by the 3-paths of $deabc$ and by $bijdcghe$, we obtain that $H^{R,B}$ contains the 3-cycle eab contradicting (3.1.2). See Figure 3.1.9m.

In all cases we obtain a contradiction which implies that G has Frank number different from 2. This finishes the proof of Theorem 3.1.35. \square

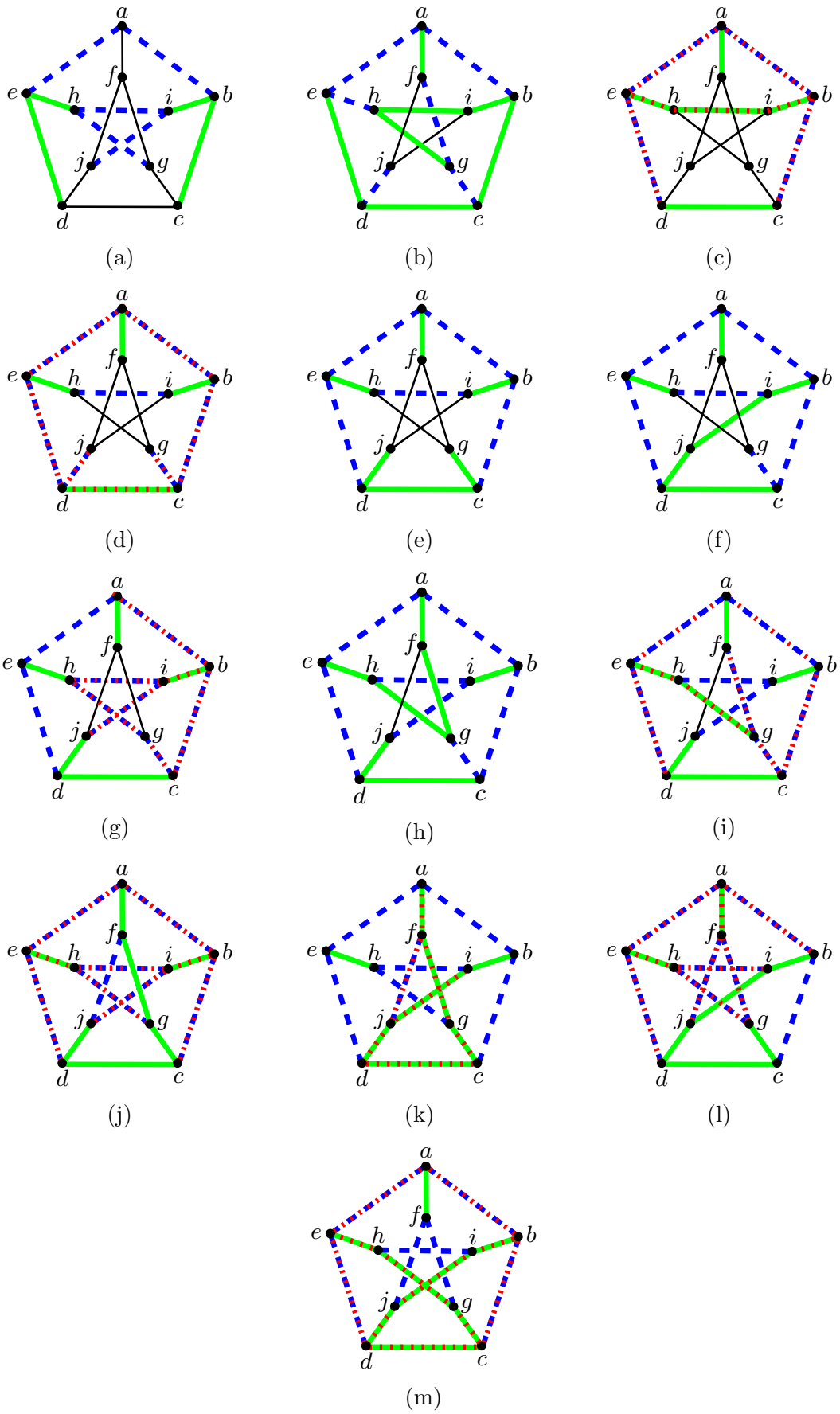


Figure 3.1.9

3.1.4.5 Algorithmic aspects

This section is dedicated to proving Theorem 3.1.36.

Our reduction is from CMNAE3-SAT, see Section 2.3.3.3. First, we introduce our construction and show that the constructed graph is cubic and 3-edge-connected. The remaining two parts are dedicated to showing that the reduction works indeed.

The construction Let $F = (\mathbf{X}, \mathcal{C})$ be a CMNAE3-SAT formula with $X = \{x_1, \dots, x_m\}$. If there is a variable $x \in X$ that is contained in only one clause $C \in \mathcal{C}$, then F is satisfiable if and only if $(X - x, \mathcal{C} - C)$ is satisfiable. We may therefore assume that every $x_i \in X$ is contained in at least 2 clauses. For $i = 1, \dots, m$, we define p_i to be the number of clauses x_i is contained in.

We now construct an instance $(G = (\mathbf{V}, \mathbf{E}), \mathbf{S})$ of DELETABILITY. For $i = 1, \dots, m$, let G contain a cycle \mathbf{K}_i of length $2p_i$. We abbreviate $V(\mathbf{K}_i)$ to \mathbf{V}_i and $E(\mathbf{K}_i)$ to \mathbf{E}_i . Observe that V_i can be partitioned into two stable sets in a unique way. We call one of these sets \mathbf{A}_i and the other one \mathbf{B}_i . Note that $|A_i| = |B_i| = p_i$. For every clause C , G contains a vertex v_C . We denote $\{v_C : C \in \mathcal{C}\}$ by $\mathbf{V}_{\mathcal{C}}$. Further, G contains a cycle \mathbf{K} of length $3|\mathcal{C}|$. We abbreviate $V(\mathbf{K})$ to $\mathbf{V}_{\mathbf{K}}$ and $E(\mathbf{K})$ to $\mathbf{E}_{\mathbf{K}}$. We add a perfect matching between $\{v_C : x_i \in C\}$ and A_i for every $i = 1, \dots, m$ and between $\bigcup_{i=1}^m B_i$ and $V_{\mathbf{K}}$. Observe that this is possible because $|A_i| = p_i$ and $|\bigcup_{i=1}^m B_i| = \sum_{i=1}^m p_i = 3|\mathcal{C}| = |V_{\mathbf{K}}|$. Finally, we define $\mathbf{S} = \bigcup_{i=1}^m \mathbf{E}_i$. Note that $|V| = 10|\mathcal{C}|$ and $|E| = 15|\mathcal{C}|$, so the construction is polynomial indeed.

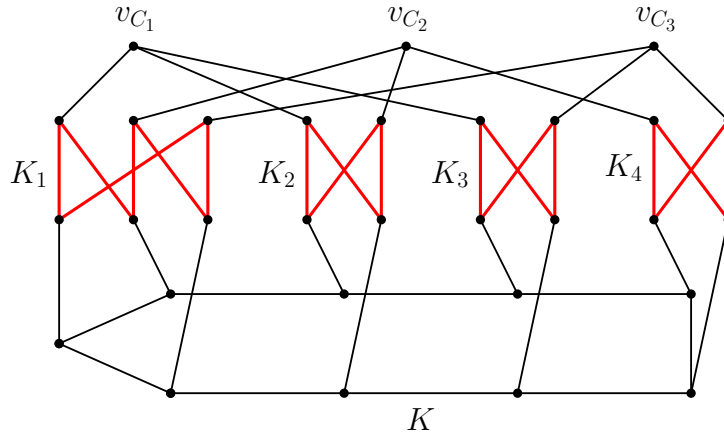


Figure 3.1.10

Figure 3.1.10 shows the constructed graph for the formula consisting of the variables x_1, \dots, x_4 and the clauses $C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_1, x_2, x_4\}$ and $C_3 = \{x_1, x_3, x_4\}$. The edges of S are marked in red.

Observe that G is cubic as every clause contains exactly 3 variables and by construction. We show that it also satisfies the other desired structural property.

Lemma 3.1.17. G is 3-edge-connected.

Proof. Assume for the sake of a contradiction that G contains some cut $\delta(Z)$ which consists of at most 2 edges. Without loss of generality, we may assume that $V_{\mathbf{K}} \cap Z$ is nonempty.

Claim 3.1.24. $V_{\mathbf{K}} \subseteq Z$.

Proof. Assume that there is a vertex $w \in V_K - Z$. As $G - V_K$ arises from G_F by replacing vertices by cycles and G_F is connected by assumption, $G - V_K$ is connected. Then, since a perfect matching exists between $\bigcup_{i=1}^m B_i$ and V_K , we obtain that $G - E_K$ is also connected. As K is 2-edge-connected, it follows that $2 \geq d_G(Z) = d_K(Z) + d_{G-E_K}(Z) \geq 2 + 1 = 3$, a contradiction. \square

Claim 3.1.25. $V_C \subseteq Z$.

Proof. Consider a vertex v_C where C contains the variables x_i, x_j, x_ℓ . By construction, both v_C and K have a neighbor in each of V_i, V_j and V_ℓ and K_i, K_j and K_ℓ are connected. As $V_K \subseteq Z$ by Claim 3.1.24, there are 3 edge-disjoint paths from v_C to Z . It follows, by $d_G(Z) \leq 2$, and Theorem 2.1.2 that $v_C \in Z$. \square

By Claims 3.1.24 and 3.1.25, there exists a vertex $v \in V_i - Z$ for some $i = 1, \dots, m$ and v is connected to $V_K \cup V_C \subseteq Z$ by a path of length 1 and two paths of length 2 and all of these are edge-disjoint. By Theorem 2.1.2, this is a contradiction to Z being separated from v by a cut of at most 2 edges. This finishes the proof of Lemma 3.1.17. \square

The remaining part of this section is dedicated to showing that our construction is indeed correct, i.e. F is a positive instance of CMNAE3-SAT if and only if (G, S) is a positive instance of DELETABILITY.

From orientation to truth assignment Suppose that (G, S) is a positive instance of DELETABILITY, so there is an orientation D of G such that $D - \vec{s}$ is strongly connected for all $s \in S$. Before finding a feasible truth assignment of the formula, we need the following result about the orientation.

Claim 3.1.26. *Let $i \in \{1, \dots, m\}$. Then all the arcs between A_i and B_i are directed in the same way.*

Proof. Let v be any vertex of K_i and e, f the two edges of K_i incident to v . Since $e, f \in S$, $D - \vec{e}$ and $D - \vec{f}$ are strongly connected. Then, as G is cubic, \vec{e} and \vec{f} are either both entering or both leaving v . Since K_i is connected, the claim follows. \square

Using Claim 3.1.26, we now define a truth assignment of X in the following way: a variable x_i is assigned the value TRUE if the arcs between A_i and B_i are directed from B_i to A_i and FALSE if the arcs between A_i and B_i are directed from A_i to B_i .

Consider a clause $C = \{x_i, x_j, x_\ell\}$. The vertex v_C has one neighbor in each of A_i, A_j and A_ℓ in G . As D is strongly connected and G is cubic, v_C has one in-neighbor w , say in A_ℓ and w has an in-neighbor in $D[V_\ell]$. It follows by construction that x_ℓ is set to TRUE in the truth assignment. Similarly, one of x_i, x_j, x_ℓ is set to FALSE. It follows that the assignment is feasible.

From truth assignment to orientation Assume that there is a feasible truth assignment for an instance F of CMNAE3-SAT consisting of a variable set $X = \{x_1, \dots, x_m\}$ and a clause set \mathcal{C} . Relabeling variables, we may assume that there is some $t \in \{0, \dots, m\}$ such that x_i is set to TRUE for $i = 1, \dots, t$ and x_i is set to FALSE for $i = t + 1, \dots, m$. Let $\mathcal{A}_1 = \bigcup_{i=1}^t A_i$, $\mathcal{A}_2 = \bigcup_{i=t+1}^m A_i$, $\mathcal{B}_1 = \bigcup_{i=1}^t B_i$ and $\mathcal{B}_2 = \bigcup_{i=t+1}^m B_i$.

We define an orientation D of G as follows. We orient all edges from P to R where P and R are two consecutive sets in $\mathcal{A}_1, V_C, \mathcal{A}_2, \mathcal{B}_2, V_K, \mathcal{B}_1, \mathcal{A}_1$. Finally, we orient the edges of K as a circuit.

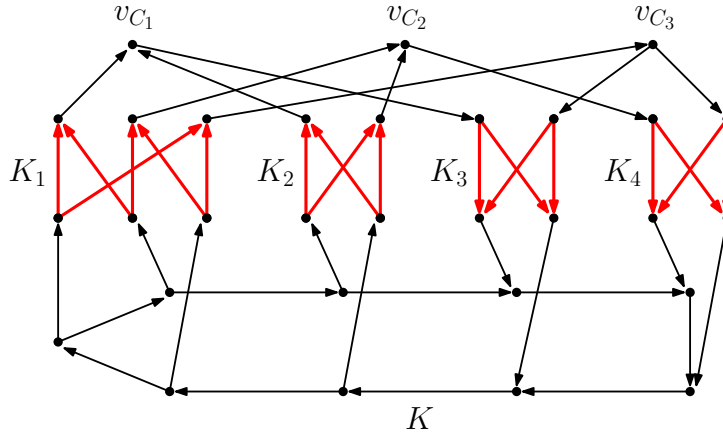


Figure 3.1.11

Figure 3.1.11 shows the obtained orientation for the formula consisting of the variables x_1, \dots, x_4 and the clauses $C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_1, x_2, x_4\}$ and $C_3 = \{x_1, x_3, x_4\}$ when x_1 and x_2 are set to TRUE and x_3 and x_4 are set to FALSE.

The following is the orientation's decisive property:

Claim 3.1.27. *In D , every vertex $v_C \in V_C$ has an in-neighbor in \mathcal{A}_1 and an out-neighbor in \mathcal{A}_2 .*

Proof. Let C contain the 3 variables x_i, x_j and x_ℓ . As the truth assignment is feasible, one of x_i, x_j, x_ℓ , say x_i , is set to TRUE and a different one, say x_j , is set to FALSE. Then, by construction, D contains an arc from $A_i \subseteq \mathcal{A}_1$ to v_C and an arc from v_C to $A_j \subseteq \mathcal{A}_2$. \square

The following result will finish the proof:

Claim 3.1.28. *Let $s \in S$. Then $D - \vec{s}$ is strongly connected.*

Proof. Since K is oriented as a circuit, all vertices of K are in the same strongly connected component Q . By construction, all vertices in \mathcal{B}_1 have an in-neighbor in $V_K \subseteq Q$ and all vertices in \mathcal{A}_1 have 2 in-neighbors in \mathcal{B}_1 in D , so at least one in $D - \vec{s}$. It follows, by Claim 3.1.27, that all vertices in $\mathcal{A}_1 \cup \mathcal{B}_1 \cup V_C$ are reachable from Q in $D - \vec{s}$. By similar arguments, Q is reachable from all vertices in $\mathcal{A}_2 \cup \mathcal{B}_2 \cup V_C$. This yields that $V_C \subseteq Q$. Finally, from every vertex in $\mathcal{A}_1 \cup \mathcal{B}_1$ there exists a directed path of length 1 or 2 to a vertex $v_C \in V_C$. Similarly, to every vertex in $\mathcal{A}_2 \cup \mathcal{B}_2$ there exists a directed path of length 1 or 2 from a vertex $v_C \in V_C$. It follows that $D - \vec{s}$ is strongly connected. \square

This reduction proves Theorem 3.1.36.

3.1.4.6 Conclusion

Our work shows that $f(G) \leq 7$ for every 3-edge-connected graph G and that $f(G) = 3$ if G is the Petersen graph. Also, we show a better bound for the more restricted classes of essentially 4-edge-connected graphs and 3-edge-colorable, 3-edge-connected graphs. Further, we show that a graph of Frank number bigger than 5 would imply the failure of Conjecture 3.1.2. Moreover, the decision problem whether all edges of a given subset can become deletable in one orientation is proven to be NP-complete.

The most obvious remaining problem is to improve these bounds on the Frank number in the general case. Considering the indications found during our work, we propose the following conjecture:

Conjecture 3.1.3. *Every 3-edge-connected graph G satisfies $f(G) \leq 3$.*

A possible way to make progress towards Conjecture 3.1.3 would be the following generalization of Lemmas 3.1.13 and 3.2.16. Using the fact that cubic graphs are 4-edge-colorable [109] and similar arguments as before, Conjecture 3.1.4 would imply that $f(G) \leq 4$ for any 3-edge-connected graph.

Conjecture 3.1.4. *Let M be a matching of a 3-edge-connected graph G intersecting each 3-edge-cut of G in at most one edge. Then M is deletable.*

It would also be interesting to generalize Frank numbers to arbitrary odd connectivity. There are two possible such generalizations. Observe that for fixed k the solution to the first problem is at least as large as the solution to the second one.

Research Problem 3.1.11. *Given a $(2k + 1)$ -edge-connected graph G , what is the minimum number of k -arc-connected orientations such that each edge becomes an arc whose deletion does not destroy k -arc-connectivity in at least one of these orientations?*

Research Problem 3.1.12. *Given a $(2k + 1)$ -edge-connected graph $G = (V, E)$, what is the minimum number of orientations such that for every $F \subseteq E$ with $|F| \leq k$, we have that $D - \vec{F}$ is strongly connected for at least one of these orientations D ?*

It follows from a theorem in [23] that these numbers are bounded by a constant depending only upon k . We are particularly interested in whether or not these numbers can be bounded by a constant not depending upon k .

3.2 Orientations for vertex-connectivity

In this section, we consider orientations that satisfy different connectivity conditions than in Section 3.1. We want the orientations to be highly vertex-connected. In Section 3.2.1, we give an overview of the results concerning the most natural analogue of the results in Section 3.1. We search for graphs admitting one orientation of high global vertex-connectivity.

In Section 3.2.2, we search for Eulerian graphs with a much stronger property: we want each of their Eulerian orientations to be highly vertex-connected. We identify some such graph classes.

3.2.1 Finding one highly vertex-connected orientation

In many aspects, the theory of orientations for vertex-connectivity is much more complicated than the theory of orientations for arc-connectivity. To start with, a good necessary condition for a graph having a k -vertex-connected orientation is not entirely obvious. The following characterization was conjectured by Frank in [42].

Conjecture 3.2.1. *A graph $G = (V, E)$ having at least $k + 1$ vertices has a k -vertex-connected orientation if and only if $G - X$ is $2(k - |X|)$ -edge-connected for every $X \subseteq V$ with $|X| \leq k$.*

Partial results for Conjecture 3.2.1 for $k = 2$ were found by Gerards [57], Berg and Jordán [11] and Cheriyan, Durand de Gevigney and Szigeti [20]. Finally, Thomassen managed to prove Conjecture 3.2.1 for $k = 2$ in [105]. While the proof of Thomassen is elementary and not very long, its writing is very dense and it actually is pretty technical. It would therefore be interesting to find a simple proof of this result.

On the negative side, Conjecture 3.2.1 was disproven by Durand de Gevigney in [24]. Two more counterexamples for $k = 3$ due to Szigeti can be found in [61]. The first one is interesting because it only contains 6 vertices and therefore is smaller than the one in [24]. The second one is interesting because it is the first counterexample not containing any double edges. The idea of the constructions comes from [24, 25].

Further, Durand de Gevigney proved in [24] that the corresponding algorithmic problem is not tractable, so a nice characterization seems out of reach.

Theorem 3.2.1. *For any $k \geq 3$, the problem of deciding whether a given graph has a k -vertex-connected orientation is NP-complete.*

It remains interesting to consider approximative versions of Conjecture 3.2.1. In particular, the following conjecture which is due to Thomassen [104] is still open.

Conjecture 3.2.2. *For every $k \in \mathbb{Z}_{\geq 0}$, there is some $f(k)$ such that every $f(k)$ -vertex-connected graph has a k -vertex-connected orientation.*

Observe that the theorem of Thomassen gives an affirmative answer to Conjecture 3.2.2 for $k = 2$, namely it shows that $f(2) = 4$. In the light of Theorem 3.2.1, there is no hope to find a good characterization of mixed graphs admitting highly vertex-connected orientations. Nevertheless, the following problem remains open. It is hinted at by Thomassen in [105] and explicitly asked by Bang-Jensen, Huang and Zhu in [7].

Research Problem 3.2.1. *Is there a polynomial time algorithm that decides whether a mixed graph has a 2-vertex-connected orientation?*

Clearly, a positive answer to Research Problem 3.2.1 should imply the theorem of Thomassen. We next show that such a result would also imply an algorithmic result that is closely related to the result on finding 2-arc-connected orientations of Frank, Király and Király in [50].

Theorem 3.2.2. *If there is a polynomial time algorithm that decides whether a given mixed graph has a 2-vertex-connected orientation, then there is also a polynomial time algorithm that decides whether a given mixed graph has a 2-arc-connected orientation.*

Proof. Let $G = (V, A \cup E)$ be a mixed graph. Clearly, if there is some $v \in V$ that is incident to at most 1 arc or edge in G , then G does not have a 2-arc-connected orientation, so we may suppose that $d_A^-(v) + d_A^+(v) + d_E(v) \geq 2$ for all $v \in V$. We create another mixed graph $H = (V', A' \cup E')$. For every $v \in V$, we let V' contain a set X_v of $d_A^-(v) + d_A^+(v) + d_E(v)$ vertices. For every $v \in V$ and for every $x, y \in X_v$, we let A' contain an arc from x to y and an arc from y to x . For every $uv \in A$, we let A' contain an arc from X_u to X_v and for every $uv \in E$, we let E' contain an edge between X_u and X_v . We do this in a way so that for every $v \in V$ and every $x \in X_v$, we have that x is incident to exactly one arc or edge in $(A' \cup E') - A'[X_v]$. An example can be found in Figure 3.2.1.

Observe that $|V'| = \sum_{v \in V} (d_A^-(v) + d_A^+(v) + d_E(v)) = 2|A \cup E|$, $|E'| = |E|$ and

$$\begin{aligned} |A'| &= \sum_{v \in V} 2 \binom{d_A^-(v) + d_A^+(v) + d_E(v)}{2} + |A| \\ &\leq \sum_{v \in V} (d_A^-(v) + d_A^+(v) + d_E(v))^2 + |A| \\ &\leq (2(|A| + |E|))^2 + |A|. \end{aligned}$$

It follows that the size of H is polynomial in the size of G . Further, by assumption, we have a polynomial time algorithm that decides whether H has a 2-vertex-connected

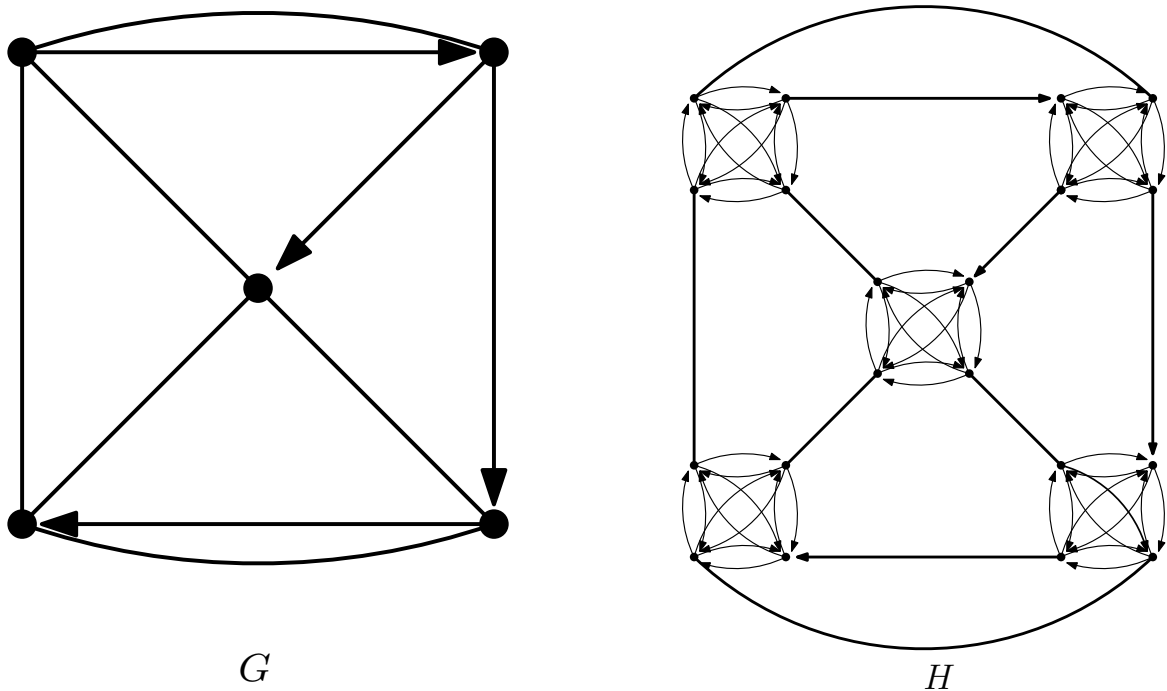


Figure 3.2.1: An example for the construction in Theorem 3.2.2.

orientation. It hence suffices to prove that H has a 2-vertex-connected orientation if and only if G has a 2-arc-connected orientation.

First suppose that H has a two-vertex-connected orientation \vec{H} . In particular, \vec{H} is 2-arc-connected. As $|X_v| \geq 2$ for all $v \in V$, we can now create an orientation \vec{G} of G by contracting X_v into v for all $v \in V$. By Proposition 2.1.2 (a), we obtain that \vec{G} is 2-arc-connected.

Now suppose that G has a 2-arc-connected orientation \vec{G} . Let an orientation \vec{H} of H be obtained from \vec{G} by replacing v by X_v and adding the arcs of $H[X_v]$ for all $v \in V$. Let $v \in V$ and $x \in X_v$. We need to show that $\vec{H} - x$ is strongly connected. As \vec{G} is 2-arc-connected, we have that $\vec{G} - a$ is strongly connected, where a is the unique arc in $A(\vec{H}) - A'(X_v)$ that is incident to x . Further, as $|X_v| \geq 2$, we have that $\vec{H}[X_v - x]$ is strongly connected and $\vec{H}[X_w]$ is strongly connected for all $w \in V' - v$. It hence follows that $\vec{H} - x$ is strongly connected by Proposition 2.1.3(b). \square

3.2.2 Graphs all of whose Eulerian orientations are highly vertex-connected

By Proposition 2.1.11, any Eulerian orientation of an Eulerian $2k$ -edge-connected graph is k -arc-connected. Clearly, the analogous statement does not hold for vertex-connectivity as can be seen by considering a triangle all of whose edges are doubled. Nevertheless, the question of finding more restricted classes of such graphs remains interesting. We focus on $2k$ -regular graphs. More concretely, we say that a $2k$ -regular graph having at least $k + 2$ vertices is *good* if each of its Eulerian orientations is k -vertex-connected, *bad* otherwise. The study of good graphs has been initiated by Levit, Chandran and Cheriyan [85] where the following result is proved.

Theorem 3.2.3. *Hypercubes are good.*

A simpler proof of Theorem 3.2.3 has been found by Szigeti, see [61]. Cheriyan asked whether more classes of good graphs can be found [19]. We here give a new, simpler proof

for hypercubes being good and provide some more classes of good graphs. This work is based on [61].

3.2.2.1 Preliminaries

We first give a preparatory result that contains a reformulation of the definition of bad graphs. It will be used frequently throughout this section.

Proposition 3.2.1. *A $2k$ -regular simple graph $G = (V, E)$ with at least $k + 1$ vertices is bad if and only if there exists an orientation D of G and a partition of V into nonempty sets Z, S and T such that*

$$d_D^-(v) = d_D^+(v) = k \text{ for all } v \in V, \quad (3.2.1)$$

$$|Z| = k - 1, \quad (3.2.2)$$

$$\text{every edge of } \delta_G(S, T) \text{ is oriented from } S \text{ to } T \text{ in } D. \quad (3.2.3)$$

Further, if an orientation D of G satisfies (3.2.1)–(3.2.3) with some partition of V into S, T and Z , it also satisfies

$$G[S] \text{ contains a cycle}, \quad (3.2.4)$$

$$d_D^-(S) \leq k \min\{|Z|, |S|\}, \quad (3.2.5)$$

$$d_G(S, T) \leq k^2 - k - i_G(Z). \quad (3.2.6)$$

Moreover, S can be chosen so that

$$|S| \leq |T|, \quad (3.2.7)$$

$$\text{every vertex } s \text{ of } S \text{ has an out-neighbor in } S \text{ in } D. \quad (3.2.8)$$

Proof. First suppose that G is bad, so there is an Eulerian orientation D of G that is not k -vertex-connected. As D is Eulerian, (3.2.1) holds. As D is not k -vertex-connected, there is a set Z of size $k - 1$ such that $D - Z$ is not strongly connected. As $D - Z$ is not strongly connected, there is a nonempty, proper subset S of $V - Z$ such that $d_{D-Z}^-(S) = 0$. Setting $T = V - Z - S$, we obtain (3.2.2) and (3.2.3).

We next show that if any orientation D satisfies (3.2.1)–(3.2.3) with a partition of V into S, T and Z , it also satisfies (3.2.4)–(3.2.6) with the same partition.

As G is simple, so is D . Together, with (3.2.2), this yields $d_D(Z, s) \leq k - 1$ for every $s \in S$. Further, by (3.2.3), we have $d_D(T, s) = 0$. Using (3.2.1), we obtain $d_{D[S]}^-(s) = d_D^-(s) - d_D(Z, s) - d_D(T, s) \geq k - (k - 1) - 0 = 1$. As s was chosen arbitrarily, we obtain that $D[S]$ contains a circuit by Proposition 2.1.6. Clearly, this yields a cycle in $G[S]$, so (3.2.4) holds. Since, by (3.2.1), $d_D^-(s) = k$ for all $s \in S$, it follows that

$$d_D^-(S) \leq \sum_{s \in S} d_D^-(s) = \sum_{s \in S} k = k|S|.$$

Moreover, by (3.2.3), all arcs entering S come from Z . Further, by (3.2.1), we have $d_D^+(z) = k$ for all $z \in Z$. This yields

$$d_D^-(S) \leq d_D^+(Z) \leq \sum_{z \in Z} d_D^+(z) = \sum_{z \in Z} k = k|Z|.$$

The last two equations yield (3.2.5).

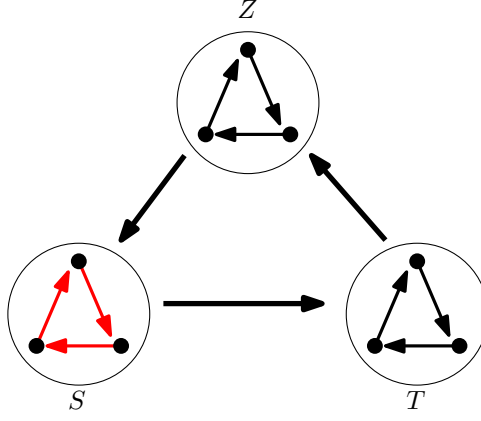


Figure 3.2.2: An example for the orientation and the partition in Proposition 3.2.1. The thick arcs between the sets indicate the orientation of all arcs between these sets. The arcs corresponding to the cycle in $G[S]$ are marked in red.

By (3.2.3), we have $d_G(S, T) \leq d_D^+(S)$. By (3.2.1) and Proposition 2.1.9, we have $d_D^+(S) = d_D^-(S)$. By (3.2.3), we have $d_D^-(S) \leq d_D^+(Z)$. Further, (3.2.2) yields $|Z| = k - 1$. We obtain

$$\begin{aligned} d_G(S, T) &\leq d_D^+(S) = d_D^-(S) \leq d_D^+(Z) = \sum_{z \in Z} (d_D^+(z) - d_{D[Z]}^+(z)) \\ &= k|Z| - i_G(Z) = k^2 - k - i_G(Z), \end{aligned}$$

so (3.2.6) holds.

In order to show (3.2.7) and (3.2.8), let us choose an orientation D of G and a partition Z, S and T of V satisfying (3.2.1) – (3.2.3) so that $|S|$ is minimum. Observe that the orientation \bar{D} that is obtained from D by reversing all arcs of D also satisfies the conditions (3.2.1)–(3.2.3) when considering the partition of V into $S' = T, T' = S$ and Z . As $|S|$ was chosen to be minimum, we have $|S| \leq |S'| = |T|$. This yields (3.2.7).

The fact that G is simple and (3.2.4) imply $|S| \geq 2$. Suppose that there is some $s \in S$ with $d_{D[S]}^+(s) = 0$. Let $S' = S - s$ and $T' = T \cup s$. Observe that D also satisfies (3.2.1)–(3.2.3) with the partition of V into S', T' and Z . As $|S'| < |S|$, this contradicts the choice of S , so (3.2.8) follows.

Now suppose that G has an orientation D satisfying (3.2.1)–(3.2.3). By (3.2.1), we have that D is Eulerian and by (3.2.2) and (3.2.3), we have that D is not k -vertex-connected. It follows by definition that G is bad. \square

An example for an orientation and a partition like in Proposition 3.2.1 for K_9 can be found in Figure 3.2.2. Unfortunately, it turns out that only very limited classes of graphs can be proven to be good. As an indication for that, we next show that almost all complete graphs are bad which rules out a big number of natural graph classes.

Theorem 3.2.4. *The complete graphs K_{2k+1} are bad for all $k \geq 4$.*

Proof. Let $k \geq 4$ be an integer and $G = (V, E)$ the complete graph K_{2k+1} . By $k \geq 4$, we have $\lfloor \frac{k}{2} \rfloor \geq 2$. This yields

$$\begin{aligned} (\lfloor \frac{k}{2} \rfloor + 1) + 2(\lceil \frac{k}{2} \rceil + 1) &= (\lfloor \frac{k}{2} \rfloor + \lceil \frac{k}{2} \rceil) + (\lceil \frac{k}{2} \rceil + 2) + 1 \\ &\leq 2(\lfloor \frac{k}{2} \rfloor + \lceil \frac{k}{2} \rceil) + 1 \\ &= 2k + 1. \end{aligned}$$

We can hence choose three disjoint sets S, T and Z' in V such that $|S| = \lfloor \frac{k}{2} \rfloor + 1$ and $|T| = |Z'| = \lceil \frac{k}{2} \rceil + 1$. Let $Z := V \setminus (S \cup T)$. Note that $Z \supseteq Z'$ and

$$|Z| = |V| - |S| - |T| = 2k + 1 - (\lfloor \frac{k}{2} \rfloor + 1) - (\lceil \frac{k}{2} \rceil + 1) = k - 1.$$

Let M be the empty set if k is even and a perfect matching of the graph $G' = (T \cup Z', \delta_G(T, Z'))$ if k is odd. Since $|T| = |Z'|$ and G is a complete graph, G' is a regular complete bipartite graph, so M exists. Let us orient all edges in $\delta_G(S, T)$ from S to T , all edges in $\delta_G(T, Z') \setminus M$ from T to Z' and all edges in $\delta_G(Z', S)$ from Z' to S . Let F be the set of edges that are oriented already and observe that $d_{\vec{F}}^+(v) = \lceil \frac{k}{2} \rceil + 1 = d_{\vec{F}}^-(v)$ for all $v \in S \cup T \cup Z'$ and $d_{\vec{F}}^+(v) = 0 = d_{\vec{F}}^-(v)$ for all $v \in Z' - Z$. It follows that \vec{F} induces an Eulerian directed graph. Hence, by Proposition 2.1.10, F induces an Eulerian subgraph of G . Since G is Eulerian, $G - F$ is also Eulerian, so $G - F$ has an Eulerian orientation by Proposition 2.1.10. Combining the orientation of F with an arbitrary Eulerian orientation of $G - F$, we have an orientation D of G and a partition $\{Z, S, T\}$ of V that satisfy (3.2.1), (3.2.2) and (3.2.3). Thus, by Proposition 3.2.1, $G = K_{2k+1}$ is bad. An illustration for K_9 can be again be found in Figure 3.2.2. \square

3.2.2.2 Classes of good graphs

In this section, we show that the following graph families are good: the complete bipartite graphs $K_{2k, 2k}$, the incidence graphs of projective planes of even degree, the line graphs of regular complete bipartite graphs, the line graphs of complete graphs and the hypercubes Q_{2k} . In the last case, this is a new proof for the theorem of Levit, Chandran and Cheriyan.

We will apply the following easy observation: for all triples of real numbers (a, b, c) with $a, b \geq c$, since $(a - c)(b - c) \geq 0$, we have

$$ab \geq c(a + b - c). \quad (3.2.9)$$

Let a be a non-negative integer. We use the notation $\binom{a}{2}$ for $\frac{a(a-1)}{2}$ and we apply the following inequality:

$$\binom{a}{2} \geq \max\{a - 1, 2a - 3\}. \quad (3.2.10)$$

3.2.2.2.1 Complete bipartite graphs

Let us first consider even regular complete bipartite graphs. For some $k \in \mathbb{Z}_{\geq 0}$, the regular complete bipartite graph $K_{2k, 2k}$ is defined by $\mathbf{V}(K_{2k, 2k}) = V_1 \cup V_2$ for two sets V_1, V_2 of size $2k$ and $\mathbf{E}(K_{2k, 2k}) = \{v_1 v_2 : v_1 \in V_1, v_2 \in V_2\}$. For an illustration, see Figure 3.2.3.

Theorem 3.2.5. *The complete bipartite graphs $K_{2k, 2k}$ are good for all $k \geq 1$.*

Proof. We assume for a contradiction that the bipartite graph $G = (V_1, V_2; E) = K_{2k, 2k}$ is bad. By Proposition 3.2.1, there exists an orientation D of G and a partition of $V_1 \cup V_2$ into nonempty sets Z, S and T such that (3.2.1) – (3.2.6) are satisfied. For $i = 1, 2$, let $\mathbf{Z}_i := Z \cap V_i$, $\mathbf{S}_i := S \cap V_i$ and $\mathbf{T}_i := T \cap V_i$ and $z_i := |Z_i|$, $s_i := |S_i|$ and $t_i := |T_i|$. Note that we have $z_1, z_2 \geq 0$ and, by (3.2.2), we have $z_1 + z_2 = |Z| = k - 1$.

Claim 3.2.1. *The following hold:*

$$s_1 + s_2 + t_1 + t_2 = 3k + 1, \quad (3.2.11)$$

$$1 \leq s_1, s_2, t_1, t_2 \leq k, \quad (3.2.12)$$

$$s_1, s_2, t_1, t_2 \in \mathbb{Z}. \quad (3.2.13)$$

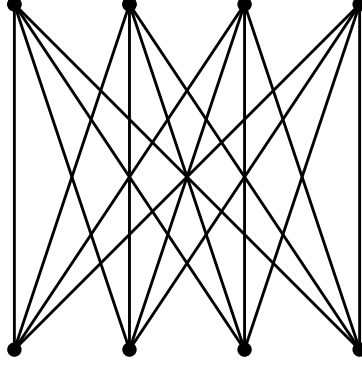


Figure 3.2.3: A drawing of the complete bipartite graph $K_{4,4}$.

Proof. By $|V(G)| = 4k$ and $|Z| = k - 1$, we have

$$s_1 + s_2 + t_1 + t_2 = |S| + |T| = |V(G)| - |Z| = 4k - (k - 1) = 3k + 1,$$

so (3.2.11) holds. By $S \neq \emptyset$, without loss of generality we may assume that there exists $v \in S \cap V_1$, so $s_1 \geq 1$. Then, by (3.2.1) and because G is bipartite and simple, v has k neighbors in V_2 . By (3.2.3), $z_1 + z_2 = k - 1$ and $z_1 \geq 0$, we obtain that $z_2 = (z_1 + z_2) - z_1 \leq (k - 1) - 0 = k - 1$, so $d_D(Z_2, v) \leq k - 1$. By (3.2.3), we have $d_D(T_2, v) = 0$. This yields, $d_D(S_2, v) = d_D^-(v) - d_D(Z_2, v) - d_D(T_2, v) \geq k - (k - 1) - 0 = 1$, so in particular $s_2 \geq 1$. By similar arguments, we obtain $t_1, t_2 \geq 1$. Moreover, by (3.2.1), (3.2.3), $v \in S \cap V_1$ and the fact G is a complete bipartite graph, we have $k = d_D^+(v) \geq d_G(v, T \cap V_2) = t_2$ and similarly $s_1, s_2, t_1 \leq k$, so (3.2.12) holds. By definition, (3.2.13) obviously holds. \square

Claim 3.2.2. *The minimum of $s_1 t_2 + s_2 t_1$ subject to (3.2.11), (3.2.12) and (3.2.13) is $k^2 + k$.*

Proof. Let the minimum be attained at $(\bar{s}_1, \bar{s}_2, \bar{t}_1, \bar{t}_2)$. First suppose that $k > \bar{s}_1 > 1$ and $k > \bar{t}_2 > 1$. By symmetry, we may suppose that $k > \bar{s}_1 \geq \bar{t}_2 > 1$. We now consider $(\bar{s}'_1, \bar{s}'_2, \bar{t}'_1, \bar{t}'_2) := (\bar{s}_1 + 1, \bar{s}_2, \bar{t}_1, \bar{t}_2 - 1)$. By (3.2.11), we have

$$\begin{aligned} \bar{s}'_1 + \bar{s}'_2 + \bar{t}'_1 + \bar{t}'_2 &= (\bar{s}_1 + 1) + \bar{s}_2 + \bar{t}_1 + (\bar{t}_2 - 1) \\ &= \bar{s}_1 + \bar{s}_2 + \bar{t}_1 + \bar{t}_2 \\ &= 3k + 1, \end{aligned}$$

so $(\bar{s}'_1, \bar{s}'_2, \bar{t}'_1, \bar{t}'_2)$ satisfies (3.2.11). Clearly, we also have $\bar{s}'_1 > \bar{s}_1 \geq 1, 1 \leq \bar{s}'_2 = \bar{s}_2 \leq k, 1 \leq \bar{t}'_1 = \bar{t}_1 \leq k$ and $\bar{t}'_2 < \bar{t}_2 \leq k$. Further, by (3.2.13), we also have $\bar{s}'_1 = \bar{s}_1 + 1 \leq (k - 1) + 1 = k$ and $\bar{t}'_2 = \bar{t}_2 - 1 \geq 2 - 1 = 1$. It follows that $(\bar{s}'_1, \bar{s}'_2, \bar{t}'_1, \bar{t}'_2)$ satisfies (3.2.12). By definition and (3.2.13), we obtain that $(\bar{s}'_1, \bar{s}'_2, \bar{t}'_1, \bar{t}'_2)$ also satisfies (3.2.13).

Further, by $\bar{s}_1 \geq \bar{t}_2$, we obtain

$$\begin{aligned} \bar{s}'_1 \bar{t}'_2 + \bar{s}'_2 \bar{t}'_1 &= (\bar{s}_1 + 1) \cdot (\bar{t}_2 - 1) + \bar{s}_2 \bar{t}_1 \\ &= \bar{s}_1 \bar{t}_2 + \bar{s}_2 \bar{t}_1 - \bar{s}_1 + \bar{t}_2 - 1 \\ &< \bar{s}_1 \bar{t}_2 + \bar{s}_2 \bar{t}_1. \end{aligned}$$

This contradicts the choice of $(\bar{s}_1, \bar{s}_2, \bar{t}_1, \bar{t}_2)$. So either $\max\{\bar{s}_1, \bar{t}_2\} = k$ or $\min\{\bar{s}_1, \bar{t}_2\} = 1$. Similarly, either $\max\{\bar{s}_2, \bar{t}_1\} = k$ or $\min\{\bar{s}_2, \bar{t}_1\} = 1$. Suppose that one of $\min\{\bar{s}_1, \bar{t}_2\}$ and $\min\{\bar{s}_2, \bar{t}_1\}$ equals 1, say $\bar{s}_1 = \min\{\bar{s}_1, \bar{t}_2\} = 1$. In this case, by (3.2.12) and (3.2.11), we obtain $3k + 1 = \bar{s}_1 + \bar{s}_2 + \bar{t}_1 + \bar{t}_2 \leq 1 + k + k + k = 3k + 1$, so equality holds throughout. This yields $s_2 = t_1 = t_2 = k$. In any case, we have $\max\{\bar{s}_1, \bar{t}_2\} = k = \max\{\bar{s}_2, \bar{t}_1\}$.

By (3.2.11), this yields

$$\begin{aligned}
\bar{s}_1\bar{t}_2 + \bar{s}_2\bar{t}_1 &= \max\{\bar{s}_1, \bar{t}_2\} \cdot \min\{\bar{s}_1, \bar{t}_2\} + \max\{\bar{s}_2, \bar{t}_1\} \cdot \min\{\bar{s}_2, \bar{t}_1\} \\
&= k(\min\{\bar{s}_1, \bar{t}_2\} + \min\{\bar{s}_2, \bar{t}_1\}) \\
&= k(3k + 1 - (\max\{\bar{s}_1, \bar{t}_2\} + \max\{\bar{s}_2, \bar{t}_1\})) \\
&= k(3k + 1 - 2k) \\
&= k^2 + k.
\end{aligned}$$

□

By Claims 3.2.1 and 3.2.2 and (3.2.6), we have $k^2 + k \leq s_1t_2 + s_2t_1 = d_G(S, T) \leq k^2 - k$. Then, by $k \geq 1$, we have a contradiction that completes the proof of Theorem 3.2.5. □

We mention that the previous proof can be easily modified to show that the bipartite graphs obtained from $K_{2k+1, 2k+1}$ by deleting a perfect matching are good for all $k \geq 1$.

3.2.2.2 Incidence graphs of projective planes

Let G be the incidence graph of a non-degenerate projective plane of order $2k - 1$. It is well-known that G is a simple connected $2k$ -regular bipartite graph with unique color classes V_1 and V_2 both being of size $(2k - 1)^2 + (2k - 1) + 1 = 4k^2 - 2k + 1$. For an exact definition of G , see [53]. For our purposes, it suffices to know the following important property of G :

$$\text{any two vertices in } V_i \text{ have exactly one common neighbor for } i \in \{1, 2\}. \quad (3.2.14)$$

Theorem 3.2.6. *The incidence graph $G = (V_1, V_2; E)$ of a projective plane of order $2k - 1$ is good for all $k \geq 1$.*

Proof. We assume for a contradiction that G is bad. Then, by Proposition 3.2.1, there exists an orientation D of G and a partition of $V_1 \cup V_2$ into nonempty sets Z, S and T such that (3.2.1) – (3.2.8) are satisfied.

For $i = 1, 2$, let $\mathbf{S}_i, \mathbf{T}_i, \mathbf{Z}_i$ be $V_i \cap S, V_i \cap T$ and $V_i \cap Z$, respectively, and let $\mathbf{s}_i := |S_i|$, $\mathbf{t}_i := |T_i|$ and $\mathbf{z}_i := |Z_i|$.

Claim 3.2.3. $s_1t_1 \leq z_2k^2 + d_G(S, T)(2k - 1)$.

Proof. For every pair $(s, t) \in S_1 \times T_1$, by (3.2.14), exactly one (s, t) -path of length 2 exists. We say that a pair $(s, t) \in S_1 \times T_1$, is of *Type 1* if the interior vertex of this path is in Z_2 and of *Type 2* if it is in $S_2 \cup T_2$. Observe that every pair $(s, t) \in S_1 \times T_1$ is of exactly one of the 2 types. We let γ_i denote the number of pairs of Type i for $i = 1, 2$.

For a vertex $z \in Z_2$, since $d_G(z, S_1) + d_G(z, T_1) \leq d_G(z) = 2k$, there are $d_G(z, S_1)d_G(z, T_1) \leq k^2$ pairs $(s, t) \in S_1 \times T_1$ such that the interior vertex of the path of length 2 between s and t is z . It follows that $\gamma_1 \leq k^2z_2$.

Observe that for every pair (s, t) of Type 2, there is an edge in $\delta_G(S, T)$ that is on the unique path of length 2 between s and t . Let $uv \in \delta_G(S, T)$ with $u \in S_1$ and $v \in T_2$. Then for every pair $(s, t) \in S_1 \times T_1$ such that uv is on the unique path of length 2 between s and t , we have $s = u$ and $t \in N_G(v) - s$. It follows that there are at most $|N_G(v)| - 1 = 2k - 1$ pairs $(s, t) \in S_1 \times T_1$ such that uv is on the unique path of length 2 between s and t . Similarly, for every edge uv with $u \in S_2$ and $v \in T_1$, there are at most $|N_G(v)| - 1 = 2k - 1$ pairs $(s, t) \in S_1 \times T_1$ such that uv is on the unique path of length 2 between s and t . This yields $\gamma_2 \leq d_G(S, T)(2k - 1)$. We obtain $s_1t_1 = \gamma_1 + \gamma_2 \leq z_2k^2 + d_G(S, T)(2k - 1)$. □

By (3.2.7), we have $s_1 + s_2 = |S| \leq |T| = t_1 + t_2$, so either $s_1 \leq t_1$ or $s_2 \leq t_2$, say $s_1 \leq t_1$.

Since G is bipartite and simple, (3.2.4) implies that $G[S]$ contains a cycle of length at least 4. As G is bipartite, this cycle must contain at least 2 vertices $v_1, v_2 \in S_2$. As G is $2k$ -regular, (3.2.1) and (3.2.3) imply $|N_G(v_i) \cap (S_1 \cup Z_1)| \geq d_D^-(v_i) \geq k$ for $i = 1, 2$. Now (3.2.14) yields

$$\begin{aligned} s_1 + z_1 &\geq |N_G(v_1) \cap (S_1 \cup Z_1)| + |N_G(v_2) \cap (S_1 \cup Z_1)| - |N_G(v_1) \cap N_G(v_2) \cap (S_1 \cup Z_1)| \\ &\geq k + k - 1 \\ &= 2k - 1. \end{aligned}$$

Then, by $z_1 \leq k - 1$ and $t_1 \geq s_1$, we have $t_1 \geq s_1 \geq 2k - 1 - z_1 \geq k$.

By (3.2.9) applied to (s_1, t_1, k) and $s_1 + t_1 + z_1 = |V_1|$, we obtain $s_1 t_1 \geq k(s_1 + t_1 - k) = k(|V_1| - z_1 - k)$. Now Claim 3.2.3, (3.2.2), (3.2.6), $|V_1| = 4k^2 - 2k + 1$ and $k \geq 1$ yield

$$\begin{aligned} k(|V_1| - z_1 - k) &\leq s_1 t_1 \\ &\leq z_2 k^2 + d_G(S, T)(2k - 1) \\ &\leq (k - 1 - z_1)k^2 + (k^2 - k)(2k - 1) \\ &= k(3k^2 - 4k + 1 - z_1 k) \\ &< k(|V_1| - k - z_1), \end{aligned}$$

a contradiction that completes the proof of Theorem 3.2.6. \square

3.2.2.2.3 Line graphs of regular complete bipartite graphs

Let us consider the regular complete bipartite graph $K_{k+1, k+1}$ and denote its bipartition classes by $\{x_1, \dots, x_{k+1}\}$ and $\{y_1, \dots, y_{k+1}\}$. This part deals with its line graph $L(K_{k+1, k+1})$: the vertex set of $L(K_{k+1, k+1})$ is the set $\{(x_i, y_j) : 1 \leq i, j \leq k + 1\}$ and two distinct vertices (x_i, y_j) and $(x_{i'}, y_{j'})$ are connected by an edge if $i = i'$ or $j = j'$. We mention that $L(K_{k+1, k+1})$ is also called *rook graph*. The graph $L(K_{k+1, k+1})$ for $k = 2$ is given in Figure 3.2.4.

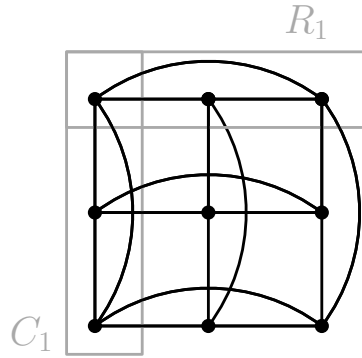


Figure 3.2.4: A drawing of $L(K_{3,3})$ with the row R_1 and the column C_1 .

Note that $L(K_{k+1, k+1})$ is $2k$ -regular.

By a *row* R_i (resp. *column* C_j) of $L(K_{k+1, k+1})$ we denote the vertex set $\{(x_i, y_j) : 1 \leq j \leq k + 1\}$ (resp. $\{(x_i, y_j) : 1 \leq i \leq k + 1\}$). The set of rows (resp. columns) is denoted by \mathcal{R} (resp. \mathcal{C}). By a *line* we mean a row or a column. The set of lines is denoted by \mathcal{L} . Observe that \mathcal{R} contains $k + 1$ rows, \mathcal{C} contains $k + 1$ columns, \mathcal{L} contains $2k + 2$ lines and every line contains $k + 1$ vertices. Note that, by construction, it follows that

- (a) each line of $L(K_{k+1, k+1})$ is a clique of $L(K_{k+1, k+1})$,

(b) a line and a stable set of $L(K_{k+1,k+1})$ have at most one vertex in common.

We need the following preparatory result that was proven in a slightly different form by König [83]

Proposition 3.2.2. *Every induced subgraph H of $L(K_{k+1,k+1})$ has a proper coloring with $\omega(H)$ colors.*

We are now ready to proceed to the main proof.

Theorem 3.2.7. *$L(K_{k+1,k+1})$ is good for all $k \geq 1$.*

Proof. Let $G = L(K_{k+1,k+1})$ for some $k \geq 1$ and assume for a contradiction that G is bad. Then, by Proposition 3.2.1, there exists an orientation D of G and a partition of $V(G)$ into nonempty sets Z, S and T such that (3.2.1) – (3.2.6) are satisfied. For a line $L_i \in \mathcal{L}$, let s_i, t_i and z_i denote $|L_i \cap S|, |L_i \cap T|$ and $|L_i \cap Z|$, respectively. Since $|L_i| = k + 1$, the following holds:

$$s_i + t_i + z_i = k + 1. \quad (3.2.15)$$

Let \mathcal{R}_S (resp. \mathcal{R}_T) be the set of rows that are disjoint from T (resp. S). The column classes \mathcal{C}_S and \mathcal{C}_T are similarly defined. Let $\mathcal{L}_S := \mathcal{R}_S \cup \mathcal{C}_S$, $\mathcal{L}_T := \mathcal{R}_T \cup \mathcal{C}_T$ and \mathcal{L}' the rest of the lines.

Note that, by definition, we have

$$\text{the intersection of a line of } \mathcal{L}_S \text{ and a line of } \mathcal{L}_T \text{ is in } Z. \quad (3.2.16)$$

In the first part of the proof we show that one of \mathcal{L}_S or \mathcal{L}_T contains at least half of the lines. We first provide a lower bound on the number of lines in $\mathcal{L}_S \cup \mathcal{L}_T$.

Claim 3.2.4. *$\mathcal{L}_S \cup \mathcal{L}_T$ contains at least $k + 2$ lines.*

Proof. Let L_i in \mathcal{L}' . By definition of \mathcal{L}' , we have $s_i, t_i \geq 1$. Hence, by (3.2.9) applied to $(s_i, t_i, 1)$, we obtain that $s_i t_i \geq 1(s_i + t_i - 1) = s_i + t_i - 1$. By (a) and (3.2.15), this yields that L_i contains at least $s_i + t_i - 1 = k - z_i$ edges between S and T .

Observe that, as every vertex belongs to exactly one row and one column and by (3.2.2), we have $\sum_{L_i \in \mathcal{L}} z_i = 2(k - 1)$.

Therefore, by (3.2.6), we have

$$(k - 1)k \geq d_G(S, T) \geq \sum_{L_i \in \mathcal{L}'} (k - z_i) \geq |\mathcal{L}'|k - 2|Z| > (|\mathcal{L}'| - 2)k,$$

thus $|\mathcal{L}'| \leq k$. Hence, by $|\mathcal{L}| = 2k + 2$, we have

$$|\mathcal{L}_S| + |\mathcal{L}_T| = |\mathcal{L}| - |\mathcal{L}'| \geq (2k + 2) - k = k + 2.$$

□

Now we show in several steps that one of \mathcal{L}_S and \mathcal{L}_T is almost empty.

Claim 3.2.5. *One of $\mathcal{R}_S, \mathcal{R}_T, \mathcal{C}_S$ and \mathcal{C}_T is empty.*

Proof. Suppose for a contradiction that none of $\mathcal{R}_S, \mathcal{R}_T, \mathcal{C}_S$ and \mathcal{C}_T are empty. Hence $|\mathcal{R}_S|, |\mathcal{C}_T| \geq 1$ and so, by (3.2.9) applied to $(|\mathcal{R}_S|, |\mathcal{C}_T|, 1)$, we obtain $|\mathcal{R}_S||\mathcal{C}_T| \geq 1(|\mathcal{R}_S| + |\mathcal{C}_T| - 1) = |\mathcal{R}_S| + |\mathcal{C}_T| - 1$. Similarly, $|\mathcal{R}_T||\mathcal{C}_S| \geq |\mathcal{R}_T| + |\mathcal{C}_S| - 1$.

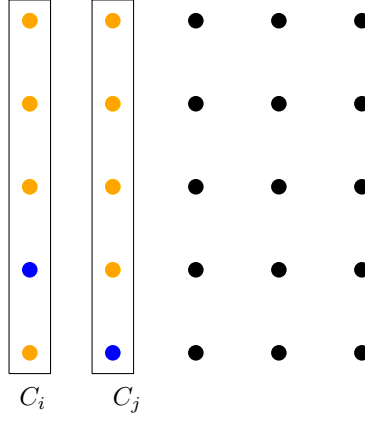


Figure 3.2.5: A schematic drawing for Claim 3.2.6. The orange vertices belong to s , the blue vertices belong to T , more exactly X . The black vertices can belong to an arbitrary class of the partition. Observe that all edges have been left out.

By Claim 3.2.4, (3.2.2) and (3.2.16), this yields

$$\begin{aligned}
|\mathcal{R}_S||\mathcal{C}_T| + |\mathcal{R}_T||\mathcal{C}_S| &\geq (|\mathcal{R}_S| + |\mathcal{C}_T| - 1) + (|\mathcal{R}_T| + |\mathcal{C}_S| - 1) \\
&= |\mathcal{L}_S| + |\mathcal{L}_T| - 2 \\
&\geq (k + 2) - 2 \\
&> |Z| \\
&\geq |\mathcal{R}_S||\mathcal{C}_T| + |\mathcal{R}_T||\mathcal{C}_S|,
\end{aligned}$$

a contradiction. □

By Claim 3.2.5, we may suppose that \mathcal{C}_S is empty. Indeed, by symmetry of G , we can exchange the rows and columns of G if needed, we may hence suppose that one of \mathcal{C}_S and \mathcal{C}_T is empty. Observe that in the digraph obtained from D by reversing all arcs the partition of $V(G)$ into Z, T and S satisfies (3.2.1), (3.2.2) and (3.2.3). Therefore, eventually exchanging the role of S and T and reversing the arcs of D , we may suppose that \mathcal{C}_S is empty.

Claim 3.2.6. *At most one column contains at least k vertices of S .*

Proof. Suppose there exist two columns C_i and C_j such that $s_i, s_j \geq k$. By $\mathcal{C}_S = \emptyset$, we have $t_i, t_j \geq 1$. Then, by (3.2.15) and $z_i \geq 0$, we have $k + 1 \geq s_i + t_i \geq k + 1$, so $s_i = k$ and $t_i = 1$. Similarly, we have $s_j = k$ and $t_j = 1$. Let \mathbf{v}_i (\mathbf{v}_j) be the unique vertex in $T \cap C_i$ ($T \cap C_j$) and let $X := \{\mathbf{v}_i, \mathbf{v}_j\}$.

Note that $|X| = 2$, $X \subseteq T$ and $(C_i \cup C_j) \setminus X \subseteq S$.

So, by (3.2.3), all the neighbors of X in C_i and C_j are in-neighbors of X , and hence all the arcs leaving X enter columns different from C_i and C_j . By (a), we obtain that $d_D^-(X) \geq d_{C_i}^-(\mathbf{v}_i) + d_{C_j}^-(\mathbf{v}_j) \geq 2k$. Further, by $|\mathcal{C}| = k + 1$, we have $d_D^+(X) \leq d_{G-(C_i \cup C_j)}^+(\mathbf{v}_i) + d_{G-(C_i \cup C_j)}^+(\mathbf{v}_j) \leq 2(k - 1)$. This yields $d_D^+(X) < d_D^-(X)$, a contradiction to (3.2.1) by Proposition 2.1.9. A schematic drawing can be found in Figure 3.2.5. □

Claim 3.2.7. \mathcal{L}_S contains at most one line.

Proof. Suppose for a contradiction that $|\mathcal{L}_S| \geq 2$. Since \mathcal{C}_S is empty, we have $|\mathcal{R}_S| \geq 2$. By Claim 3.2.6, at most one column C_i satisfies $s_i \geq k$.

Now consider some column $C_j \neq C_i$. We have $s_j + z_j \geq |\mathcal{R}_S| \geq 2$. Further, as $j \neq i$, by (3.2.15), we have $t_j + z_j = (k + 1) - s_j \geq (k + 1) - (k - 1) = 2$.

Applying (3.2.9) to $(s_j, t_j, 2 - z_j)$, we hence obtain by (3.2.15) that $s_j t_j \geq (2 - z_j)(s_j + t_j - (2 - z_j)) = (2 - z_j)(k - 1)$. By (a), we get that every column $C_j \in \mathcal{C}' := \mathcal{C} \setminus (\mathcal{C}_T \cup \{C_i\})$ contains at least $(2 - z_j)(k - 1)$ edges between S and T .

By (3.2.16), the columns in \mathcal{C}_T contain at least $|\mathcal{R}_S||\mathcal{C}_T|$ vertices of Z . Observe that by $|\mathcal{C}| = k + 1$, we have $|\mathcal{C}'| = |\mathcal{C} - C_i| - |\mathcal{C}_T| = k - |\mathcal{C}_T|$. Also, note that, by (3.2.2) and as \mathcal{C}' and \mathcal{C}_T are disjoint, we have

$$\begin{aligned} k - 1 &= |Z| \\ &\geq \sum_{C_j \in \mathcal{C}'} z_j + \sum_{C_j \in \mathcal{C}_T} z_j \\ &\geq \sum_{C_j \in \mathcal{C}'} z_j + |\mathcal{R}_S||\mathcal{C}_T|. \end{aligned}$$

Then, by (3.2.6), since the $G[C_j]$'s are edge-disjoint and $|\mathcal{R}_S| \geq 2$, we have

$$\begin{aligned} (k - 1)k &\geq d_G(S, T) \\ &\geq \sum_{C_j \in \mathcal{C}'} d_{G[C_j]}(S, T) \\ &\geq \sum_{C_j \in \mathcal{C}'} (2 - z_j)(k - 1) \\ &\geq (k - 1)(2(k - |\mathcal{C}_T|) - ((k - 1) - |\mathcal{R}_S||\mathcal{C}_T|)) \\ &> (k - 1)(k + (|\mathcal{R}_S| - 2)|\mathcal{C}_T|) \\ &\geq (k - 1)k, \end{aligned}$$

a contradiction. □

We can now see that \mathcal{L}_T contains at least half of the lines. Indeed, Claims 3.2.4 and 3.2.7 imply that

Claim 3.2.8. \mathcal{L}_T contains at least $k + 1$ lines.

In the second part of the proof our goal is to give an upper bound on the size of S . In order to do that we consider a particular vertex-coloring of $\mathbf{H} := G[S]$. By Proposition 3.2.2, there exists a proper vertex-coloring \mathcal{I} of H with $\omega(H)$ colors.

Claim 3.2.9. S contains at most $2\omega(H) - 1$ vertices.

Proof. Let U be the set of vertices in the lines of \mathcal{L}_T , $Z' = Z \cap U$ and $Z'' = Z \setminus Z'$. Let I be a color class in \mathcal{I} . Since I is a stable set in S , by (b), each vertex in U has at most one neighbor in I and each vertex of Z'' has at most two neighbors in I . Since every inneighbor of a vertex in S either belongs to S or to Z , we obtain

$$d_D^-(S) = \sum_{I \in \mathcal{I}} |\delta_D^-(S) \cap \delta_D^-(I)| \leq \sum_{I \in \mathcal{I}} (|Z'| + 2|Z''|) = \omega(H)(|Z'| + 2|Z''|). \quad (3.2.17)$$

Let v be a vertex in a color class $I \in \mathcal{I}$. It follows, by (a) and Claim 3.2.8, that v has at least $|\mathcal{L}_T| \geq k + 1$ neighbors in U . So I has at least $|I|(k + 1)$ neighbors in U , each being, by (3.2.3), either a vertex in Z' or an outneighbor of v in D . Hence

$$d_D^+(S) = \sum_{I \in \mathcal{I}} |\delta_D^+(S) \cap \delta_D^+(I)| \geq \sum_{I \in \mathcal{I}} (|I|(k + 1) - |Z'|) = |S|(k + 1) - \omega(H)|Z'|. \quad (3.2.18)$$

Then, (3.2.18), (3.2.1), Proposition 2.1.9, (3.2.17), $Z' \cup Z'' = Z$ and (3.2.2) yield that

$$\begin{aligned}
|S| &\leq \lfloor \frac{d_D^+(S) + \omega(H)|Z'|}{k+1} \rfloor \\
&= \lfloor \frac{d_D^-(S) + \omega(H)|Z'|}{k+1} \rfloor \\
&\leq \lfloor \frac{\omega(H)(|Z'| + 2|Z''|) + \omega(H)|Z'|}{k+1} \rfloor \\
&= \lfloor \frac{2\omega(H)(|Z'| + |Z''|)}{k+1} \rfloor \\
&= \lfloor \frac{2\omega(H)(k-1)}{k+1} \rfloor \\
&\leq 2\omega(H) - 1.
\end{aligned}$$

A schematic drawing can be found in Figure 3.2.6. □

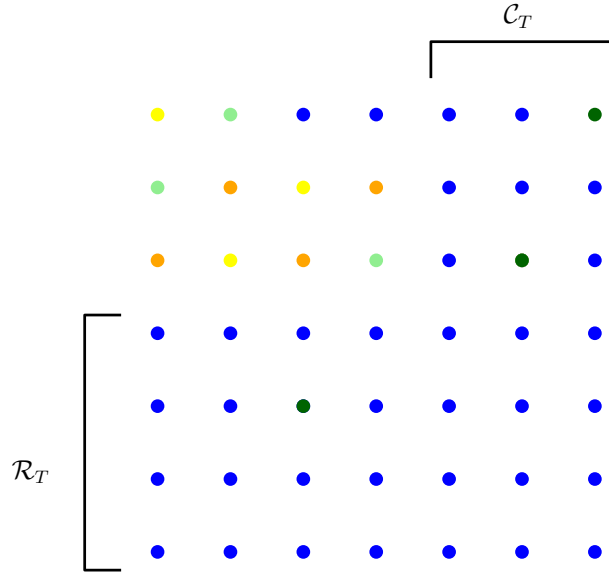


Figure 3.2.6: A schematic drawing for Claim 3.2.9. The vertices of T are marked in blue, the vertices of Z' are marked in dark green and the vertices of Z'' are marked in light green. An independent set I of $G[S]$ has been marked in yellow, while the remaining vertices of S are marked in orange. Again, all edges have been left out.

Since each clique of G is contained in a line, we can choose a line L_i that contains $\omega(H)$ vertices of S . Note that $s_i \geq 1$. Let $S_i := L_i \cap S$ and $S'_i := S \setminus S_i$.

Finally, in order to derive a contradiction, we provide bounds for $d_G(S, T)$ and $d_G(S, Z)$.

Claim 3.2.10. $s_i t_i + k s_i - (|Z| - z_i) + |S'_i| \leq d_G(S, T)$.

Proof. By (a), we have $s_i t_i = d_G(S_i, T \cap L_i)$. Next observe that every element of S_i has k neighbors which are not in L_i and these neighborhoods are disjoint. As at most $|Z \setminus L_i| + |S'_i|$ of these vertices are in $Z \cup S$, we obtain that at least $k s_i - (|Z \setminus L_i| + |S'_i|)$ of them are in T . By (a), this yields that $k s_i - (|Z| - z_i) - |S'_i| \leq d_G(S_i, T \setminus L_i)$. Now consider a vertex $v \in S'_i$. By (a) and Claim 3.2.8, v has at least $k + 1$ neighbors in $V(\mathcal{L}_T)$. As by definition, none of them is in S , at least $k + 1 - |Z| = (k + 1) - (k - 1) = 2$ are in T .

This yields $2|S'_i| \leq d_G(S'_i, T)$. By $d_G(S_i, T \cap L_i) + d_G(S_i, T \setminus L_i) + d_G(S'_i, T) = d_G(S, T)$, the claim follows. □

Claim 3.2.11. $d_G(S, Z) \leq s_i|Z| + |S'_i|$.

Proof. By (a), we have $s_i z_i = d_G(S_i, Z \cap L_i)$. Every element of $Z \setminus L_i$ has, by $S_i \subseteq L_i$, at most one neighbor in S_i and clearly at most $|S'_i|$ in S'_i . This gives, by Claim 3.2.9 and $\omega(H) = s_i$, that $d_G(Z \setminus L_i, S) \leq (|S'_i| + 1)(|Z| - z_i) \leq s_i(|Z| - z_i)$. Since $S'_i \cap L_i = \emptyset$, every element of S'_i has at most one neighbor in $L_i \cap Z$ and hence $d_G(Z \cap L_i, S'_i) \leq |S'_i|$. By $d_G(S_i, Z \cap L_i) + d_G(S, Z \setminus L_i) + d_G(S'_i, Z \cap L_i) = d_G(S, Z)$, the claim follows. \square

Now we are ready to conclude. Claim 3.2.10, (3.2.3), (3.2.1), Proposition 2.1.9 and Claim 3.2.11 yield that

$$\begin{aligned} s_i t_i + k s_i - (|Z| - z_i) + |S'_i| &\leq d_G(S, T) \\ &\leq d_D^+(S) \\ &= d_D^-(S) \\ &\leq d_G(S, Z) \\ &\leq s_i |Z| + |S'_i|. \end{aligned}$$

Then, by (3.2.15), (3.2.2), $t_i \geq 0$ and $s_i \geq 1$, we have

$$\begin{aligned} 0 &\geq s_i t_i + k s_i - (|Z| - z_i) + |S'_i| - (s_i |Z| + |S'_i|) \\ &\geq s_i t_i + s_i(k - |Z|) - (|Z| - z_i) \\ &= s_i t_i + s_i - (s_i + t_i - 2) \\ &= t_i(s_i - 1) + 2 \\ &\geq 2, \end{aligned}$$

a contradiction. This completes the proof of Theorem 3.2.7. \square

3.2.2.2.4 Line graphs of complete graphs

Let us consider the complete graph K_{k+2} and denote its vertex set by U . This part deals with its line graph $L(K_{k+2})$. Note that a pair of adjacent (resp. non-adjacent) edges in K_{k+2} corresponds to a pair of adjacent (resp. non-adjacent) vertices in $L(K_{k+2})$. Since each edge of K_{k+2} is adjacent to exactly $2k$ other edges, $L(K_{k+2})$ is $2k$ -regular. An illustration can be found in Figure 3.2.7.

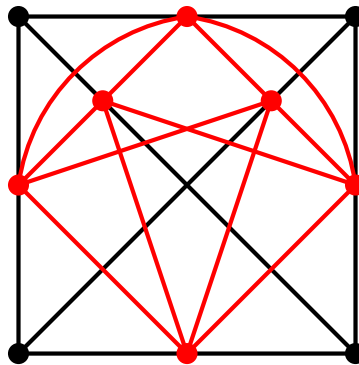


Figure 3.2.7: An illustration for the construction of the line graph. The graph K_4 is depicted in black and its line graph is depicted in red.

Theorem 3.2.8. $L(K_{k+2})$ is good for all $k \geq 1$.

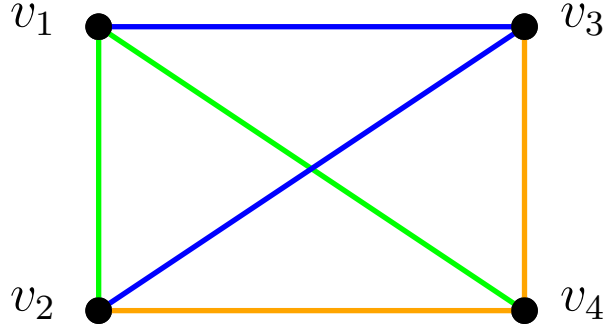


Figure 3.2.8: This is a drawing of K_4 with a marking of its edge set that corresponds to the partition of $V(G)$ into S, T and Z . The edges corresponding to vertices in S are marked in orange, the edges corresponding two vertices in T are marked in blue and the edges corresponding to vertices in Z are marked in green. The vertices corresponding to the edges v_2v_4 and v_2v_3 form a pair in P_1 while the vertices corresponding to the edges v_2v_4 and v_1v_3 form a pair in P_2 .

Proof. Let $\mathbf{G} = L(K_{k+2})$ for some $k \geq 1$ and suppose for a contradiction that G is bad. Clearly, $k \geq 2$. Then, by Proposition 3.2.1, there exists an orientation D of G and a partition of $V(G)$ into nonempty sets Z, S and T such that (3.2.1) – (3.2.8) are satisfied.

For a vertex set X of G , we denote by \mathbf{E}_X the corresponding edge set of K_{k+2} . For a vertex $v \in U$, let \mathbf{s}_v , \mathbf{t}_v and \mathbf{z}_v be the number of edges incident to v that are in E_S, E_T and E_Z , respectively. We call an ordered pair (e, f) of edges of K_{k+2} an (S, T) -pair if $e \in E_S$ and $f \in E_T$. The sets of adjacent and non-adjacent (S, T) -pairs are denoted by \mathbf{P}_1 and \mathbf{P}_2 , respectively. Observe that $|\mathbf{P}_1| = d_G(S, T)$ and $|S||T| = |\mathbf{P}_1| + |\mathbf{P}_2|$. An illustration can be found in Figure 3.2.8. First we provide an upper bound on $|\mathbf{P}_1|$.

Claim 3.2.12. $|\mathbf{P}_1| \leq k^2 - k - \max\{0, k - 4\}$.

Proof. Note that every pair of edges in E_Z which shares a vertex v in K_{k+2} provides an edge in $G[Z]$. It follows that a vertex $v \in U$ provides exactly $\binom{z_v}{2}$ edges in $G[Z]$. Then, as every such pair shares exactly one vertex in K_{k+2} , by (3.2.10) and (3.2.2), we have $i_G(Z) = \sum_{v \in U} \binom{z_v}{2} \geq \sum_{v \in U} (z_v - 1) = 2|E_Z| - |U| = 2(k - 1) - (k + 2) = k - 4$. Also, clearly, we have $i_G(Z) \geq 0$. Thus, by (3.2.6), we have $|\mathbf{P}_1| = d_G(S, T) \leq k^2 - k - i_G(Z) \leq k^2 - k - \max\{0, k - 4\}$. \square

We next prove an upper bound on $|\mathbf{P}_2|$.

Claim 3.2.13. $2|\mathbf{P}_2| \leq (k - 1)|\mathbf{P}_1| + k^2 - 3k + 2$.

Proof. A 4-cycle of K_{k+2} is called *special* if it contains a non-adjacent (S, T) -pair. Let \mathcal{C} be the set of special cycles. A special cycle is said to be of *type* i if it contains i edges of E_Z for $i = 0, 1, 2$. Let \mathbf{n}_i denote the number of special cycles of type i for $i = 0, 1, 2$.

Note that every special cycle of type 1 or 2 contains exactly one non-adjacent (S, T) -pair and every special cycle of type 0 contains at most 2 non-adjacent (S, T) -pairs. An illustration can be found in Figure 3.2.9. Further, every non-adjacent (S, T) -pair can be completed to a 4-cycle in two different ways, so every non-adjacent (S, T) -pair is part of exactly 2 special cycles. It follows that

$$2|\mathbf{P}_2| = \sum_{p \in \mathbf{P}_2} \sum_{\substack{C \in \mathcal{C} \\ p \subseteq E(C)}} 1 = \sum_{C \in \mathcal{C}} \sum_{\substack{p \in \mathbf{P}_2 \\ p \subseteq E(C)}} 1 \leq 2n_0 + n_1 + n_2. \quad (3.2.19)$$

Observe that every special cycle of type i contains $2 - i$ adjacent (S, T) -pairs for $i = 0, 1, 2$.

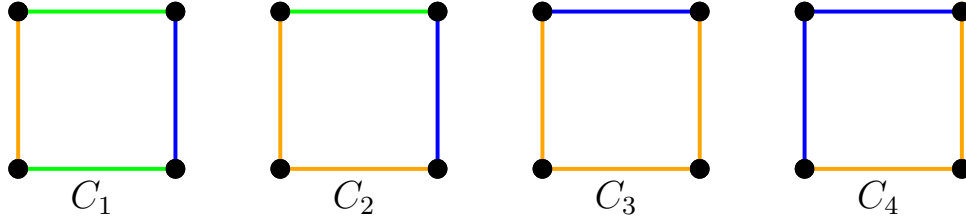


Figure 3.2.9: This drawing shows several special cycles. The edges corresponding to vertices in S are marked in orange, the edges corresponding to vertices in T are marked in blue and the edges corresponding to vertices in Z are marked in green. The cycle C_1 is a special cycle of type 2, C_2 is a special cycle of type 1, C_3 is a special cycle of type 0 that contains one non-adjacent (S, T) -pair and C_4 is a special cycle of type 0 that contains two non-adjacent (S, T) -pairs.

Also every adjacent (S, T) -pair can be completed to a 4-cycle by adding one of $k - 1$ vertices, so every adjacent (S, T) -pair is contained in exactly $(k - 1)$ 4-cycles. This yields

$$2n_0 + n_1 = \sum_{C \in \mathcal{C}} \sum_{\substack{p \in P_1 \\ p \not\subseteq E(C)}} 1 = \sum_{p \in P_1} \sum_{\substack{C \in \mathcal{C} \\ p \subseteq E(C)}} 1 = \sum_{p \in P_1} (k - 1) = (k - 1)|P_1|. \quad (3.2.20)$$

Next observe that every special cycle of type 2 contains 2 non-adjacent edges of E_Z , every pair of non-adjacent edges is contained in exactly two 4-cycles and there are at most $\binom{k-1}{2}$ pairs of non-adjacent edges of E_Z . This implies that

$$n_2 \leq 2 \binom{k-1}{2} = k^2 - 3k + 2. \quad (3.2.21)$$

The inequalities (3.2.19), (3.2.20) and (3.2.21) imply the claim. \square

We use the previous results to show an upper bound on $|S|$.

Claim 3.2.14. $|S| \leq k$.

Proof. Otherwise, by (3.2.7), we have $|T| \geq |S| \geq k + 1$. By (3.2.2), we have $|S| + |T| = |V(G)| - |Z| = \binom{k+2}{2} - (k - 1)$. Then, by (3.2.9) applied to $(|S|, |T|, k + 1)$, we have $|S||T| \geq (k + 1)(|S||T| - (k + 1)) = (k + 1)(\binom{k+2}{2} - 2k) = \frac{k^3 + k + 2}{2}$. Then Claims 3.2.13 and 3.2.12 and $k \geq 1$ yield

$$\begin{aligned} k^3 + k &\leq 2|S||T| - 2 \\ &= 2|P_2| + 2|P_1| - 2 \\ &\leq (k + 1)|P_1| + k^2 - 3k \\ &\leq (k + 1)(k^2 - k - \max\{0, k - 4\}) + k^2 - 3k \\ &= k^3 + k - (5k - k^2) - \max\{0, k^2 - 3k - 4\} \\ &= k^3 + k - \max\{k(5 - k), 2(k - 2)\} \\ &< k^3 + k, \end{aligned}$$

a contradiction. \square

The following result shows that the edges of E_S are adjacent to many edges of $E_{S \cup Z}$.

Claim 3.2.15. For every $uv \in E_S$, $s_u + z_u + s_v + z_v \geq k + 3$.

Proof. By (3.2.1), (3.2.3) and (3.2.8), the vertex of D that corresponds to uv has k neighbors in $S \cup Z$ and at least one out-neighbor in S in D and their corresponding edges in K_{k+2} are incident to u or v . As uv is counted in s_u and s_v , we obtain that $s_u + z_u + s_v + z_v \geq k + 3$. \square

The next result shows that S forms a clique in G .

Claim 3.2.16. *The edges of E_S are pairwise adjacent.*

Proof. Suppose that E_S contains two non-adjacent edges v_1v_2 and v_3v_4 . Note that K_{k+2} has 6 edges having both ends in $\{v_1, v_2, v_3, v_4\}$. Observe that an edge in $E_S \cup E_Z$ contributes 2 to $\sum_{i=1}^4 (s_{v_i} + z_{v_i})$ if it belongs to these 6 edges and 1 otherwise. Applying Claim 3.2.15 to both v_1v_2 and v_3v_4 and using Claim 3.2.14 and (3.2.2), we obtain $2(k+3) \leq \sum_{i=1}^4 (s_{v_i} + z_{v_i}) \leq |E_S| + |E_Z| + 6 \leq 2k + 5$, a contradiction. \square

Claim 3.2.17. *The edges of E_S do not form a triangle in K_{k+2} .*

Proof. Suppose that E_S forms a triangle on v_1, v_2, v_3 in K_{k+2} . Observe that every edge in E_Z is incident to at most one of v_1, v_2, v_3 and every edge in E_S is incident to exactly two of v_1, v_2, v_3 . Applying Claim 3.2.15 to all 3 edges of E_S , we get

$$\begin{aligned} 3(k+3) &\leq \sum_{uv \in E_S} (s_u + z_u + s_v + z_v) \\ &= 2 \sum_{i=1}^3 (s_{v_i} + z_{v_i}) \\ &\leq 2(2|E_S| + |E_Z|) \\ &\leq 2(6 + (k-1)), \end{aligned}$$

that contradicts $k \geq 2$. \square

By Claims 3.2.16 and 3.2.17, the edges of E_S are all incident to a vertex v in K_{k+2} . Let Q be the clique of size $k+1$ in G that corresponds to the set of edges incident to v in K_{k+2} . Note that $|S| = |Q \cap S| = s_v$, $|Q \cap T| = t_v$ and $|Q \cap Z| = z_v$. Since every edge of E_Z that is not incident to v is adjacent to at most 2 edges of E_S in K_{k+2} , each vertex of $Z \setminus Q$ is adjacent to at most 2 vertices of S in G . This implies, by (3.2.3), that $d_D^-(S) \leq d_G(S, Z \setminus Q) + d_G(S, Z \setminus Q) \leq 2|Z \setminus Q| + s_v z_v$. By (3.2.4), we have $|S| \geq 2$. Then, by (3.2.1), $s_v = |S| \geq 2$, (3.2.2), $G[S]$ is a clique, $|Q| = k+1$ and (3.2.10), we have

$$\begin{aligned} 0 &= \sum_{u \in S} (d_D^-(u) - k) \\ &= d_D^-(S) + \binom{|S|}{2} - |S|k \\ &\leq 2|Z \setminus Q| + s_v z_v + \binom{s_v}{2} - s_v(s_v - 1 + t_v + z_v) \\ &\leq 2(k-1-z_v) - 2t_v - \binom{s_v}{2} \\ &= 2(s_v - 2) - \binom{s_v}{2} \\ &< 0, \end{aligned}$$

a contradiction. This completes the proof of Theorem 3.2.8. \square

3.2.2.3 Conclusion

We provided four classes of good graphs in this section. Further investigations could allow the identification of more classes of good graphs. We are particularly interested in the graph class described below which extends two classes of good graphs.

Let W be a set of size w . The Hamming graph $H(d, w)$ is the graph with vertex set W^d , where two vertices are adjacent if they differ in exactly one coordinate. Note that $H(1, w)$ is the complete graph K_w , $H(d, 2)$ is the hypercube of dimension d and $H(2, w)$ is the line graph of $K_{w,w}$. It is easy to see that $H(d, w)$ is $d(w-1)$ -regular. We conjecture that $H(d, w)$ is a good graph whenever $d(w-1)$ is even and $d \geq 2$. This would generalize Theorems 3.2.7 and 3.2.3. We also wish to mention the following problem which has been brought to our attention by Bessy [15].

Research Problem 3.2.2. *Is there a polynomial time algorithm that decides whether a given $2k$ -regular graph is good?*

Chapter 4

Arborescence packing

4.1 Introduction to arborescence packing

The main object we deal with in this section is arborescences. Arborescences are a well-studied object in combinatorial optimization. For example, arborescences play a major role in determining the structure of shortest paths in a given rooted digraph, see Corollary 3.1.10 in [41]. Problems on covering the arc set of a given rooted digraph by spanning arborescences have been considered by Vidyasankar in [108].

The problems we are interested in in this section belong to a different class: the packing of arborescences. In the most basic setting, we are given a singularly rooted digraph $D = (V \cup r, A)$ and want to find a fixed number of arc-disjoint spanning r -arborescences in it. We call such a set of arc-disjoint arborescences a *packing* of arborescences. In 1973, Edmonds proved the following fundamental theorem on packing spanning arborescences in [28]. It gives a complete characterization of singularly rooted digraphs admitting a packing of spanning r -arborescences and it is the starting point of all research on arborescence packings.

Theorem 4.1.1. *Let $D = (V \cup r, A)$ be a singularly rooted digraph and $k \in \mathbb{Z}_{\geq 0}$. There exists a packing $\{B_1, \dots, B_k\}$ of spanning r -arborescences in D if and only if $d_A^-(X) \geq k$ for all $\emptyset \neq X \subseteq V$.*

There are several proofs of Theorem 4.1.1. Apart from the original one of Edmonds, we wish to mention the ones of Mader [90] and Lovász [88].

Further, since the appearance of Theorem 4.1.1, an enormous amount of generalizations has been found. In this section, we will discuss some of these generalizations.

As a first such generalization we consider the setting in which instead of searching for spanning arborescences all having the same root, for every arborescence, we prescribe one particular vertex to be the root of it. Theorem 4.1.1 easily allows to conclude a characterization for this more general case. The phenomenon that a more general result can be obtained from a less general one via a reduction occurs several times in this section. For the sake of an illustration, we give this theorem and its proof completely.

Theorem 4.1.2. *([28]) Let $D = (V \cup R, A)$ be a simply rooted digraph. Then there exists a packing of spanning r -arborescences $\{B_r\}_{r \in R}$ in D if and only if for all $X \subseteq V \cup R$ with $X - R \neq \emptyset$, we have*

$$d_A^-(X) \geq |R - X|. \tag{4.1.1}$$

Proof. For the necessity, let $\{B_r\}_{r \in R}$ be a packing of spanning r -arborescences in D and consider some $X \subseteq V \cup R$ with $X - R \neq \emptyset$. For every $r \in R - X$, there must be an arc in B_r that enters X . As all the B_r are arc-disjoint, we obtain $d_A^-(X) \geq |R - X|$.

For the sufficiency, let $D' = (V \cup r, A')$ be obtained by contracting all vertices of R into a single vertex r . For every $X \subseteq V$, we have $d_{A'}^-(X) = d_A^-(X) \geq |R - X| = |R|$. Hence, by Theorem 4.1.1, we obtain that D' contains a packing of $|R|$ arc-disjoint r -arborescences. As every $r \in R$ has exactly one outgoing arc in D , this yields the desired packing in D . \square

A further generalization of Theorem 4.1.1 that has also been proven by Edmonds in [28] is the following one where the simplicity condition on the rooted digraph has been omitted in comparison to Theorem 4.1.2.

Theorem 4.1.3. *Let $D = (V \cup R, A)$ be a rooted digraph. Then there exists a spanning arborescence packing $\{B_r\}_{r \in R}$ in D if and only if every $X \subseteq V \cup R$ with $X - R \neq \emptyset$ satisfies (4.1.1).*

It is interesting to remark that Theorem 4.1.3 is in some way an exception to the rule mentioned above. The only proof we know of Theorem 4.1.3 is self-contained and somewhat more technical than the proof of Theorem 4.1.1. It would be interesting to find a way to conclude Theorem 4.1.3 from Theorem 4.1.1.

In the remaining part of Section 4.1, we give an overview of previous generalizations of Theorem 4.1.1.

In Subsections 4.1.1 and 4.1.2, we consider generalizations of Theorem 4.1.1 to more general objects than digraphs, namely directed hypergraphs and mixed graphs, respectively.

In Subsection 4.1.3, we consider a much more general condition on the arborescence packing than all of the arborescences being spanning, namely matroid-based arborescence packings.

In Subsection 4.1.4, we discuss another connection between arborescence packings and matroid theory, namely a way of modeling the arc set of a packing of arborescences as the intersection of two matroids.

In Subsection 4.1.5, we discuss algorithmic aspects of arborescence packings.

After, we discuss some new generalizations of Theorem 4.1.1.

In Section 4.2, we consider a problem in which the conditions of the arborescences being spanning is somewhat relaxed. Given a digraph $D = (V, A)$ and some $r \in V$, a *reachability* r -arborescence is an r -arborescence that spans all the vertices which are reachable from r in D . We deal with several problems concerning the packing of reachability arborescences.

In Section 4.3, we deal with a problem on packing spanning arborescences where the condition of the roots of the arborescences being fixed is relaxed. Given a digraph $D = (V, A)$, a positive integer $k \in \mathbb{Z}_{\geq 0}$ and two functions $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$, we want to find a packing of k arborescences such that every $v \in V$ is the root of at least $f(v)$ and at most $g(v)$ of them. Again, we deal with this problem in several different settings.

In Section 4.4, we study packings of spanning arborescences with fixed roots where a further condition on the arborescences is imposed. More concretely, we require that in each of the arborescences, when deleting some vertex different from the root, there is still a certain number of vertices reachable from the root. Taking this number as a parameter, we prove that the problem is FPT. We show similar results for two related objects.

Before going into the details, we wish to point out that this survey on arborescence packings is far from exhaustive. For example, there are theorems on the packing of infinite arborescences ([1], [70]), on the packing of arborescences that only have to span a given subset of the vertices [4] and results on vertex-independent spanning arborescences ([66],[65],[112]).

4.1.1 Packing of spanning dyperarborescences

We consider the problem of packing dyperarborescences in directed hypergraphs. The following literal generalization of Theorem 4.1.2 was proven by Frank, Király and Kriesell in [51].

Theorem 4.1.4. *Let $\mathcal{D} = (V \cup r, \mathcal{A})$ be a singularly rooted dypergraph and $k \in \mathbb{Z}_{\geq 0}$. There exists a packing $\{B_1, \dots, B_k\}$ of spanning r -dyperarborescences in \mathcal{D} if and only if $d_{\mathcal{A}}^-(X) \geq k$ for all $\emptyset \neq X \subseteq V$.*

Clearly, when every $a \in \mathcal{A}$ has exactly one tail, we obtain Theorem 4.1.1, so Theorem 4.1.4 is more general indeed. While the original proof in [51] is based on trimming, an elegant method to conclude Theorem 4.1.4 from Theorem 4.1.1 has been found by Fortier et. al in [38].

We can conclude the following statement from Theorem 4.1.4

Theorem 4.1.5. *Let $\mathcal{D} = (V, \mathcal{A})$ be a dypergraph and R a multiset in V of size k . Some $\mathcal{A}' \subseteq \mathcal{A}$ is the dyperedge set of a packing $\{\mathcal{B}_r : r \in R\}$ of spanning r -hyperarborescences in \mathcal{D} if and only if the underlying hyperedge set of \mathcal{A}' is the hyperedge set of a packing of k spanning hypertrees and $d_{\mathcal{A}'}^-(v) = k - |R \cap v|$ for all $v \in V$.*

Proof. First suppose that \mathcal{A}' is the dyperedge set of a packing $\{\mathcal{B}_r : r \in R\}$ of spanning r -hyperarborescences in \mathcal{D} . The underlying hyperedge set of every arborescence forms a spanning hypertree, so \mathcal{A}' forms the hyperedge set of a packing of k spanning hypertrees. Further, for every $v \in V$ and each of the arborescences, the arborescence contains a dyperedge entering v if and only if v is not the root of the arborescence. It follows that $d_{\mathcal{A}'}^-(v) = k - |R \cap v|$.

Now suppose that \mathcal{A}' is the hyperedge set of a packing of k spanning hypertrees and $d_{\mathcal{A}'}^-(v) = k - |R \cap v|$ for all $v \in V$ and consider the dypergraph $\mathcal{D}' = (V, \mathcal{A}')$. Let $X \subseteq V$. As the underlying hypergraph of \mathcal{D}' is the union of k hyperedge-disjoint hypertrees, there are at most $k(|X| - 1)$ dyperedges in \mathcal{A}' whose head and tail are completely contained in X . It follows from Proposition 2.1.4 that $d_{\mathcal{A}'}^-(X) \geq \sum_{v \in X} d_{\mathcal{A}'}^-(v) - k(|X| - 1) = k|X| - |R \cap X| - k(|X| - 1) = k - |R \cap X|$. It hence follows from Theorem 4.1.4 that \mathcal{D}' contains a packing of dyperarborescences $\{\mathcal{B}_r : r \in R\}$. As this packing contains $k(|V| - 1)$ dyperedges, the statement follows. \square

4.1.2 Packing of mixed arborescences

We consider the problem of packing mixed arborescences in mixed graphs. The following characterization was proven by Frank in [44].

Theorem 4.1.6. *Let $F = (V \cup r, A \cup E)$ be a singularly rooted mixed graph and $k \in \mathbb{Z}_{\geq 0}$. Then there exists a packing $\{B_1, \dots, B_k\}$ of spanning mixed r -arborescences in F if and only if*

$$e_E(\mathcal{P}) \geq \sum_{i=1}^q (k - d_A^-(V_i))$$

holds for every subpartition $\mathcal{P} = \{V_1, \dots, V_q\}$ of V .

It is not difficult to see that if $E = \emptyset$, Theorem 4.1.6 is equivalent to Theorem 4.1.1. Further, if $A = \emptyset$, Theorem 4.1.6 reduces to Theorem 2.1.5. Hence Theorem 4.1.6 is a common generalization of Theorems 4.1.1 and Theorem 2.1.5.

4.1.3 Matroid-based packings

In this section, we discuss arborescence packings that have a more general property than all of the arborescences in it being spanning: matroid-based packings. Given a matroid-rooted digraph $(D = (V \cup R, A), M)$, an arborescence packing $\{B_r : r \in R\}$ is called *matroid-based* if $\{r \in R : v \in V(B_r)\}$ is a basis of M for all $v \in V$. The study of matroid-based packings is relatively recent, the following characterization has been found by Durand de Gevigney, Nguyen and Szigeti in 2013 [26].

Theorem 4.1.7. *Let $(D = (V \cup R, A), M = (R, r_M))$ be a simply matroid-rooted digraph. Then there exists a matroid-based arborescence packing in (D, M) if and only if for all nonempty $X \subseteq V \cup R$ with $X \cap R = \text{span}_M(N_D^-(X \cap V))$, we have*

$$d_A^-(X) \geq r_M(R) - r_M(X \cap R). \quad (4.1.2)$$

Observe Theorem 4.1.7 yields Theorem 4.1.2 if M is the free matroid on R , so Theorem 4.1.7 is a generalization of Theorem 4.1.1 indeed. Further interesting problems can be created when matroid-based packings are combined with the more general settings studied in Subsections 4.1.1 and 4.1.2. Given a matroid-rooted mixed hypergraph $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}))$, a packing of mixed hyperarborescences $\{\mathcal{B}_r : r \in R\}$ is called *matroid-based* if for every $r \in R$, there is a trimming B_r of \mathcal{B}_r such that $\{B_r : r \in R\}$ is a matroid-based packing of arborescences. These questions have been investigated in [38]. In particular, the following theorem is proven. It is an example for the appearance of bisets in arborescence packings. It is an interesting question whether a deep connection between bisets and arborescence packings can generally be established.

Theorem 4.1.8. *([38]) Let $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}), M = (R, r_M))$ be a simply matroid-rooted mixed hypergraph. Then there exists a matroid-based mixed hyperarborescence packing in (\mathcal{F}, M) if and only if for every biset subpartition $\{\mathbf{X}^i\}_1^\ell$ of V with $w(\mathbf{X}^i) = \text{span}_M(\{r \in R : N_{\mathcal{F}}^+(r) \cap X_I^i \neq \emptyset\})$ for $i = 1, \dots, \ell$, we have*

$$e_{\mathcal{E}}(\{\mathbf{X}_I^i\}_1^\ell) + \sum_{i=1}^{\ell} d_{\mathcal{A}}^-(\mathbf{X}^i) \geq \sum_{i=1}^{\ell} (r_M(R) - r_M(w(\mathbf{X}^i))). \quad (4.1.3)$$

4.1.4 Arborescence packings and matroid intersection

This subsection deals with another connection between arborescence packings and matroid theory. This connection has algorithmic consequences which will be discussed in Subsection 4.1.5. For the basic setting, given a singularly rooted digraph $D = (V \cup r, A)$ and some $k \in \mathbb{Z}_{\geq 0}$, we now define two matroids on A . Let the matroid M_1 be the k -sum of the graphic matroid of the underlying graph of D and let M_2 be the matroid that is obtained as the direct sum of the $(k, d_D^-(v))$ -uniform matroids on $\delta_D^-(v)$ for all $v \in V$. The following fruitful connection was discovered by Edmonds [31].

Theorem 4.1.9. *The arc sets of the packings of k spanning r -arborescences in D are exactly the common independent sets of M_1 and M_2 of size $k|V|$.*

We next mention a result that extends Theorem 4.1.9 to a more general setting. It is due to Király, Szigeti and Tanigawa [79].

Theorem 4.1.10. *Let $(D = (V \cup R, A), M = (R, r_M))$ be a simply matroid-rooted digraph. Then there exist two matroids M_1, M_2 on A such that the arc sets of the M -based arborescence packings of arborescences in D are exactly the common independent sets of M_1 and M_2 of size $r_M(R)|V|$. Further, independence oracles for M_1 and M_2 are available.*

We now extend this result to mixed hypergraphs.

4.1.4.1 Matroid intersection for matroid-based packings of mixed hypergraphs

This section is dedicated to modeling the hyper- and dyperedge sets of matroid-based packing of mixed hyperarborescences as the intersection of two matroids. This extends a method from [79].

We first review a slight extension of a result of [79] characterizing matroid-based packings of hyperarborescences in simply rooted dypergraphs as the intersection of two matroids. We then conclude a generalization to matroid-based packings of mixed hyperarborescences from it.

The following result was proven in [79] for the digraphic case in a slightly different form. Its proof can be literally generalized to the case of dypergraphs.

Theorem 4.1.11. *Let (\mathcal{D}, M) be a matroid-rooted dypergraph. Then the dyperedge sets of the matroid-based packings of (\mathcal{D}, M) are exactly the common independent sets of size $r(M)|V|$ of the Tanigawa matroid of (\mathcal{D}, M) and the $r(M)$ -entering matroid of \mathcal{D} .*

Let $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}), M)$ be a simply matroid-rooted mixed hypergraph and let $\mathcal{D}_{\mathcal{F}} = (V \cup R, \mathcal{A} \cup \mathcal{A}_{\mathcal{E}})$ be the directed extension of \mathcal{F} . We say that a matroid-based packing of hyperarborescences in $(\mathcal{D}_{\mathcal{F}}, M)$ is *feasible* if it contains at most one dyperedge of the bundle \mathcal{A}_e for all $e \in \mathcal{E}$. It is easy to see that (\mathcal{F}, M) has a matroid-based packing of mixed hyperarborescences if and only if $(\mathcal{D}_{\mathcal{F}}, M)$ has a feasible matroid-based packing of hyperarborescences. We are now ready to give the desired characterization. Let M_1 be the Tanigawa matroid of (\mathcal{F}, M) and let M_2 be the $r_M(R)$ -entering matroid of $\mathcal{D}_{\mathcal{F}}$.

Theorem 4.1.12. *The dyperedge sets of the feasible matroid-based packings of hyperarborescences in $(\mathcal{D}_{\mathcal{F}}, M)$ are exactly the common independent sets of size $r(M)|V|$ of M_1 and M_2 .*

Proof. First let \mathcal{A}' be the dyperedge set of a feasible matroid-based packing of hyperarborescences in $(\mathcal{D}_{\mathcal{F}}, M)$. It follows from Theorem 4.1.11 that \mathcal{A}' is a common independent set of size $r(M)|V|$ of $\mathcal{T}_{(\mathcal{D}_{\mathcal{F}}, M)}$ and M_2 . As \mathcal{A}' is feasible, \mathcal{A}' contains at most one dyperedge of the bundle \mathcal{A}_e for all $e \in \mathcal{E}$. It follows that \mathcal{A}' is also independent in M_1 . As $r(M_1) \leq r(\mathcal{T}_{(\mathcal{D}_{\mathcal{F}}, M)})$, the statement follows.

Now let \mathcal{A}' be a common independent set of size $r(M)|V|$ of M_1 and M_2 . As $r(\mathcal{T}_{(\mathcal{D}_{\mathcal{F}}, M)}) = r(M_2)$, it follows that \mathcal{A}' is also an independent set of size $r(M)|V|$ of $\mathcal{T}_{(\mathcal{D}_{\mathcal{F}}, M)}$. It follows from Theorem 4.1.11 that \mathcal{A}' is the dyperedge set of a matroid-based packing of hyperarborescences in $(\mathcal{D}_{\mathcal{F}}, M)$. As \mathcal{A}' is independent in M_1 , \mathcal{A}' contains at most one dyperedge of the bundle \mathcal{A}_e for all $e \in \mathcal{E}$ and so \mathcal{A}' is feasible. \square

4.1.5 Algorithmic aspects

In this section, we discuss algorithmic problems related to arborescence packings. There are two important related algorithmic problems. Firstly, we may want to find some arborescence packing in a certain setting. Secondly, we may want to find a minimum cost arborescence packing with respect to a given cost function. For both questions, a subtlety that has to be taken care of is that we do not only wish to find the arc set of the desired arborescence packing but also the actual arborescences. On the other hand, if in a certain setting we have an algorithm that finds an arbitrary packing of arborescences and we also have an algorithm that yields the arc set of a packing of minimum weight, we can also find a packing of minimum weight. This can be obtained by applying the first algorithm to the result of the second one. For the first problem, in most settings polynomial time algorithms can be obtained straight from the corresponding proofs. For

example, the following result is an immediate consequence of Lovasz' proof of Theorem 4.1.1 in [88].

Theorem 4.1.13. *Let $D = (V \cup r, A)$ be a singularly rooted digraph and $k \in \mathbb{Z}_{\geq 0}$. Then a packing of k spanning r -arborescences can be found in polynomial time if such a packing exists.*

The following similar result in a much more general setting was given by Fortier et. al in [38].

Theorem 4.1.14. *Let (\mathcal{F}, M) be a simply matroid-rooted mixed hypergraph. Then a matroid-based packing of mixed hyperarborescences in (\mathcal{F}, M) can be found in polynomial time if such a packing exists.*

For the weighted problem, in the basic setting, we are given a singularly rooted digraph $D = (V \cup r, A)$, an integer $k \in \mathbb{Z}_{\geq 0}$ and a weight function on A . Our aim is to find a packing of k spanning r -arborescences whose arc set is of minimum total weight. Due to Theorem 2.4.2, matroid intersection serves as a valuable tool for these weighted problems. For example, the following result is an immediate consequence of Theorems 2.4.2 and 4.1.9.

Theorem 4.1.15. *Let $D = (V \cup r, A)$ be a singularly rooted digraph, $k \in \mathbb{Z}_{\geq 0}$ and $c : A \rightarrow \mathbb{R}$ a weight function. Then a packing of spanning r -arborescences $\mathcal{B} = \{B_1, \dots, B_k\}$ minimizing $c(\mathcal{B})$ can be found in polynomial time if such a packing exists.*

Similarly, the following theorem can be obtained by combining Lemma 2.4.6, Theorem 2.4.2 and Theorem 4.1.12.

Theorem 4.1.16. *Let $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}), M)$ be a simply matroid-rooted mixed hypergraph and $c : \mathcal{A} \cup \mathcal{E} \rightarrow \mathbb{R}$ a weight function. Then a matroid-based packing of mixed hyperarborescences \mathcal{B} in (\mathcal{F}, M) minimizing $c(\mathcal{B})$ can be found in polynomial time if such a packing exists.*

We acknowledge that there is a different proof of Theorem 4.1.16 by Király[77].

4.2 Reachability in arborescence packings

In this section, we deal with a problem that is more general than packing spanning arborescences, namely packing reachability arborescences. Unless specified otherwise, all the results in this chapter can be found in [63].

Given a rooted digraph $D = (V \cup R, A)$ for which there is some $r \in R$ and some $v \in V$ such that v is not reachable from r , the only information we obtain from Theorem 4.1.2 is that no packing of spanning arborescences $\{B_r\}_{r \in R}$ exists. This raises the question whether a statement crediting the connectivity properties of D in a finer way can be made when adapting the desired properties of the arborescences in the packing. This motivated Kamiyama, Katoh and Takizawa to introduce the notion of reachability arborescences in 2009 [71]. Given a rooted digraph $D = (V \cup R, A)$ and some $r \in R$, a *reachability r -arborescence* is an r -arborescence that spans all the vertices reachable from r in D . For an illustration, see Figure 4.2.1.

It turns out that the problem of packing reachability arborescences can easily be dealt with in singularly rooted digraphs. It suffices to delete all vertices that are not reachable from the root and to apply Theorem 4.1.1. The problem becomes much more interesting when dealing with rooted digraphs with bigger root sets, so the set of vertices that needs to be spanned is not the same for every arborescence. More concretely, given a rooted

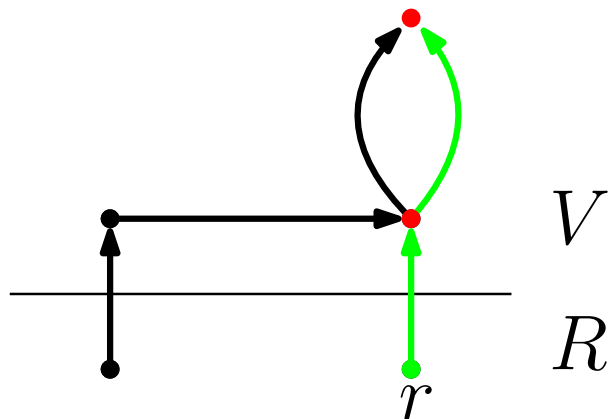


Figure 4.2.1: This is a drawing of a rooted digraph $D = (V \cup R, A)$ where the vertices which are reachable from r are marked in red and the arcs of an r -reachability arborescence are marked in green.

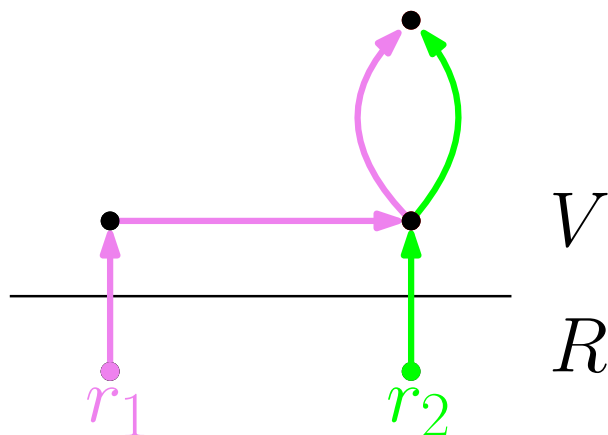


Figure 4.2.2: This is a drawing of a rooted digraph $D = (V \cup \{r_1, r_2\}, A)$ with a packing of reachability arborescences. The arcs in the reachability r_1 -arborescence are marked in violet and the arcs of the r_2 -reachability arborescence are marked in green.

digraph $D = (V \cup R, A)$, we wish to determine whether there is a packing of reachability r -arborescences $\{B_r\}_{r \in R}$. Often, we abbreviate a packing of reachability r -arborescences $\{B_r\}_{r \in R}$ to a *reachability arborescence packing* $\{B_r\}_{r \in R}$. For an illustration, see Figure 4.2.2.

Our main contribution is a new proof technique that allows to conclude some theorems on packing reachability arborescences from the corresponding theorems on spanning arborescences.

In Section 4.2.1, we show how, using Theorem 4.1.3, this technique allows to give a new proof for a theorem of Kamiyama, Katoh and Takizawa which is fundamental to the packing of reachability arborescences. In Section 4.2.2, we consider a problem combining reachability arborescences and matroid-based packings. Our proof technique allows to conclude a theorem of Király from Theorem 4.1.7. In Section 4.2.3, we consider a further generalization to mixed hypergraphs. Again, our proof technique can be applied. Finally, in Section 4.2.4, we deal with the algorithmic aspects of our results.

4.2.1 Basic setting

We here deal with the following result of Kamiyama, Katoh and Takizawa which is fundamental to the theory of reachability arborescence packings.

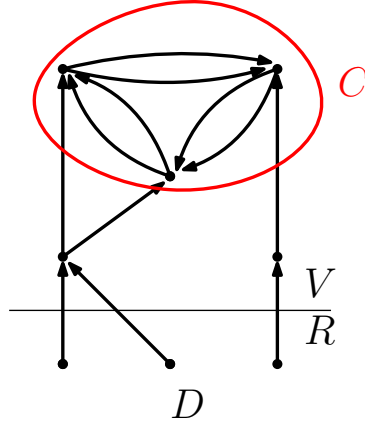


Figure 4.2.3: This is a drawing of a rooted digraph $D = (V \cup R, A)$ with the set C .

Theorem 4.2.1. ([71]) *Let $D = (V \cup R, A)$ be a rooted digraph. Then there exists a reachability arborescence packing $\{B_r\}_{r \in R}$ in D if and only if for all $X \subseteq V \cup R$ with $X \cap V \neq \emptyset$, we have*

$$d_A^-(X) \geq |P_X^D \cap R| - |X \cap R|. \quad (4.2.1)$$

Observe that if every $v \in V$ is reachable from every $r \in R$, then $P_X^D \cap R = R$ for all $X \subseteq V \cup R$ with $X \cap V \neq \emptyset$. It follows that Theorem 4.2.1 generalizes Theorem 4.1.2 indeed. The original proof of Theorem 4.2.1 in [71] is based on submodular optimization and pretty involved. A somewhat simplified version by Frank can be found in [41]. We here give a new, rather simple proof which uses Theorem 4.1.3 and is self-contained otherwise.

Proof. (of Theorem 4.2.1) For the necessity, suppose that there is some reachability arborescence packing $\{B_r\}_{r \in R}$ in D and consider some $X \subseteq V \cup R$ with $X \cap V \neq \emptyset$. For every $r \in (P_X^D \cap R) - (X \cap R)$, we obtain that B_r needs to contain some arc entering X . As all the B_r are arc-disjoint, the statement follows.

For sufficiency, let $D = (V \cup R, A)$ be a minimum counterexample. Obviously, $V \neq \emptyset$.

By Proposition 2.1.1 and as V has no arc leaving in D , there is a strongly connected component of $D[V]$ which has no arc leaving in D . We denote its vertex set by C . Note that each vertex of C is reachable in D from the same set of roots since $D[C]$ is strongly connected. An example can be found in Figure 4.2.3. We can hence divide the problem into two subproblems, a smaller one on reachability arborescence packing and one on spanning arborescence packing.

Let $D_1 = (V_1 \cup R, A_1) = D - C$. Note that D_1 is a rooted digraph.

Lemma 4.2.1. D_1 has a reachability arborescence packing $\{B_r^1\}_{r \in R}$.

Proof. By $d_A^+(C) = 0$, we have $d_{A_1}^-(X) = d_A^-(X)$ and $P_X^{D_1} = P_X^D$ for all $X \subseteq V_1 \cup R$. Then, since D satisfies (4.2.1), so does D_1 . Hence, by the minimality of D , the desired packing exists in D_1 . \square

We now create an auxiliary digraph $D_2 = (V_2 \cup R_2, A_2)$. First let $V_2 = C \cup T$ where $T = \{t_{uv} : uv \in \delta_A^-(C)\}$ is a set of new vertices and $R_2 = P_C^D \cap R$. Next, let A_2 contain $A(D[C]) \cup \{rt_{uv} : r \in R_2, u \in U_r^D, t_{uv} \in T\}$. Finally, for all $t_{uv} \in T$, let A_2 contain a set of $|R_2|$ arcs from v to t_{uv} and one arc from t_{uv} to v . An illustration can be found in Figure 4.2.4.

Lemma 4.2.2. D_2 has a spanning arborescence packing $\{B_r^2\}_{r \in R_2}$.

Proof. We show in the following claim that D_2 satisfies (4.1.1).

Claim 4.2.1. $d_{A_2}^-(X) \geq |R_2 - X|$ for all $X \subseteq V_2 \cup R_2$ with $X - R_2 \neq \emptyset$.

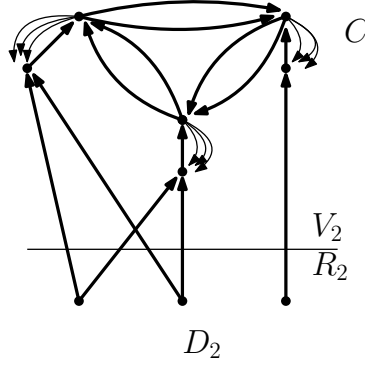


Figure 4.2.4: An illustration for the construction of D_2 from D .

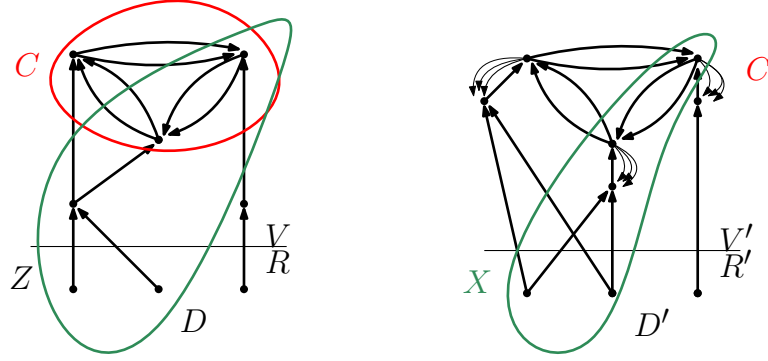


Figure 4.2.5: An illustration of the sets X and Z .

Proof. If $X \cap C = \emptyset$, we obtain from $X - R_2 \neq \emptyset$ that there is some $t_{uv} \in X$. Further, we have $v \in C \subseteq V_2 - X$. By construction, this yields $d_{A_2}^-(X) \geq d_{A_2}(v, t_{uv}) = |R_2| \geq |R_2 - X|$.

If $X \cap C \neq \emptyset$, then, since $D[C]$ is strongly connected, we have $R_2 = P_C^D \cap R = P_{X \cap C}^D \cap R$.

In order to obtain a lower bound on $d_{A_2}^-(X)$, we consider a vertex set Z in D and show that there is an injection from the arcs entering Z in D to the arcs entering X in D_2 . Let $Y = (V \cup R) - U_{R_2 - X}^D$, $Z = (X \cap C) \cup Y$ and $uv \in \delta_A^-(Z)$. For an illustration, see Figure 4.2.5.

Since, by the definition of Y , we have $\delta_A^-(Y) = \emptyset$, we obtain $v \in X \cap C$. If $u \in C$, then $uv \in \delta_{A_2}^-(X)$. If $u \notin C$, then $u \in U_{R_2 - X}^D$, so there is some $r \in R_2 - X$ such that $u \in U_r^D$. Further, as $uv \in \delta_A^-(C)$, we know that T contains the vertex t_{uv} and A_2 contains $t_{uv}v$. Also, as $u \in U_r^D$, we have that A_2 contains the arc rt_{uv} . Since $v \in X$ and $r \notin X$, we obtain that one of rt_{uv} or $t_{uv}v$ is in $\delta_{A_2}^-(X)$. Thus, by (4.2.1), $Z \subseteq P_Z^D$, $P_Z^D \cap R_2 = P_C^D \cap R_2 = R_2$ and the definition of Z , we obtain

$$\begin{aligned}
d_{A_2}^-(X) &\geq d_A^-(Z) \\
&\geq |(P_Z^D - Z) \cap R| \\
&= |(P_Z^D - Z) \cap R_2| + |(P_Z^D - Z) \cap (R - R_2)| \\
&= |P_Z^D \cap R_2| - |Z \cap R_2| + |P_Z^D \cap (R - R_2)| - |Z \cap (R - R_2)| \\
&= |R_2| - |X \cap R_2| + |R - R_2| - |R - R_2| \\
&= |R_2 - X|.
\end{aligned}$$

□

By Claim 4.2.1 and Theorem 4.1.3, the desired packing exists in D_2 . This completes the proof of Lemma 4.2.2. □

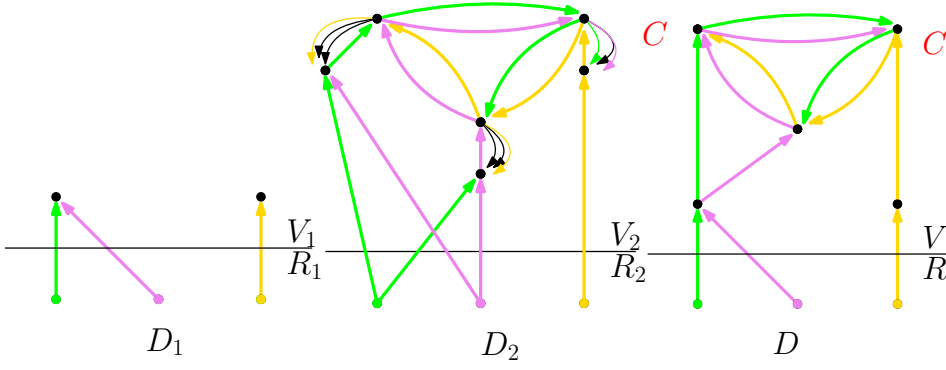


Figure 4.2.6: An example of how the packing in D is obtained from merging the packings in D_1 and D_2 .

With the help of the packings $\{B_r^1\}_{r \in R}$ in D_1 and $\{B_r^2\}_{r \in R_2}$ in D_2 obtained in Lemmas 4.2.1 and 4.2.2, a packing in D can be constructed yielding a contradiction. For an illustration, see Figure 4.2.6.

Lemma 4.2.3. *D has a reachability arborescence packing.*

Proof. For all $r \in R - R_2$, let $B_r = B_r^1$ and for all $r \in R_2$, let B_r be obtained from the union of B_r^1 and $B_r^2 - (R_2 \cup T)$ by adding the arc uv for all $t_{uv}v \in A(B_r^2)$. Since $\{B_r^1\}_{r \in R}$ and $\{B_r^2\}_{r \in R_2}$ are packings, so is $\{B_r\}_{r \in R}$. For $r \in R - R_2$, $B_r = B_r^1$ is an r -arborescence and it spans $U_r^{D_1} = U_r^D$. Let now $r \in R_2$. Since B_r^1 and B_r^2 do not contain circuits, neither does B_r . Observe that for all $v \in V(B_r^1) - r$, we have $d_{A(B_r^1)}^-(v) = 1$. Further, for all $v \in C$, we have $d_{A(B_r^2)}^-(v) = 1$, so, as $t_{uv}v \in A(B_r^2)$ is replaced by $uv \in A(B_r)$, we also have $d_{A(B_r)}^-(v) = 1$. It follows that B_r is an r -arborescence. Since B_r^1 spans $U_r^{D_1}$ and B_r^2 spans $V_2 \cup r$, we obtain that B_r spans $U_r^{D_1} \cup C = U_r^D$. It follows that $\{B_r\}_{r \in R}$ has the desired properties. This completes the proof of Lemma 4.2.3. \square

Lemma 4.2.3 contradicts the fact that D is a counterexample and hence the proof of Theorem 4.2.1 is complete. \square

4.2.2 Matroid-reachability-based arborescence packings

In this section, we consider a generalization of Theorem 4.2.1 to the matroid-based setting. Given a matroid-rooted digraph $(D = (V \cup R, A), M)$, we say that a packing of r -arborescences $\{B_r\}_{r \in R}$ is *matroid-reachability-based* if $\{r \in R : v \in V(B_r)\}$ is a basis of $P_v^D \cap R$ in M for all $v \in V$. The following characterization of matroid-rooted digraphs admitting a matroid-reachability-based arborescence packing was given by Király [76].

Theorem 4.2.2. ([76]) *Let $(D = (V \cup R, A), M = (R, r_M))$ be a matroid-rooted digraph. Then there exists a matroid-reachability-based arborescence packing in (D, M) if and only if for all $X \subseteq V \cup R$ with $X \cap V \neq \emptyset$, we have*

$$d_A^-(X) \geq r_M(P_X^D \cap R) - r_M(X \cap R). \quad (4.2.2)$$

Observe that if M is the free matroid, then Theorem 4.2.2 reduces to Theorem 4.2.1 and if every $v \in V$ is reachable from every $r \in R$, then Theorem 4.2.2 reduces to Theorem 4.1.7, so Theorem 4.2.2 generalizes both Theorem 4.2.1 and Theorem 4.1.7. The original proof of Király does not use any other theorem on arborescence packings. Again, we wish to apply our proof technique to conclude Theorem 4.2.2 from the corresponding theorem that does not include the reachability property which is Theorem 4.1.7. In order to do so,

we first need the following slight strengthening of Theorem 4.1.7 in which the simplicity condition is omitted.

Theorem 4.2.3. *Let $(D = (V \cup R, A), M = (R, r_M))$ be a matroid-rooted digraph. Then there exists a matroid-based arborescence packing in (D, M) if and only if every $X \subseteq V \cup R$ with $X \cap V \neq \emptyset$ and $X \cap R = \text{span}_M(N_D^-(X \cap V) \cap R)$ satisfies (4.1.2).*

$$d_A^-(X) \geq r_M(R) - r_M(X \cap R). \quad (4.2.3)$$

The relationship between Theorem 4.2.3 and Theorem 4.1.7 is the same as between Theorem 4.1.2 and Theorem 4.1.3. Interestingly, while no way to conclude Theorem 4.1.3 from Theorem 4.1.2 is known, the matroid setting allows to conclude Theorem 4.2.3 from Theorem 4.1.7 in a pretty simple manner.

Proof. (of **Theorem 4.2.3**) Suppose that (D, M) admits a matroid-based arborescence packing $\{\mathbf{B}_r\}_{r \in R}$ and consider some $X \subseteq V \cup R$ with $X \cap V \neq \emptyset$. Let $v \in X \cap V$ and $\mathbf{R}_v = \{r \in R : v \in V(B_r)\}$. By definition of matroid-based packings, R_v is a basis of M , so $|R_v| = r_M(R)$ by Proposition 2.4.2. Further, as R_v is independent in M , we have $|R_v \cap X| \leq r_M(X \cap R)$. It follows that $|R_v - X| \geq r_M(R) - r_M(X \cap R)$. For every $r \in R_v - X$, the arborescence B_r contains an arc entering X . As the B_r are arc-disjoint, the statement follows.

For sufficiency, let $(D' = (V \cup R', A'), M' = (R', r_{M'}))$ be the simply matroid-rooted digraph obtained from (D, M) by replacing every root $r \in R$ by a set Q_r of $|N_D^+(r)|$ simple roots in the digraph such that $N_{D'}^+(Q_r) = N_D^+(r)$ and by $|Q_r|$ parallel copies of r in the matroid.

Now let $X' \subseteq V \cup R'$ with $X' \cap R' = \text{span}_{M'}(N_{D'}^-(X' \cap V) \cap R')$. Observe that for every $r \in R$, either $Q_r \subseteq X'$ or $Q_r \cap X' = \emptyset$. Let $X = (X' \cap V) \cup \{r \in R : Q_r \subseteq X'\}$. Observe that $X \cap R = \text{span}_M(N_D^-(X \cap V) \cap R)$. Further, we have $d_A^-(X) = d_{A'}^-(X')$, $r_M(R) = r_{M'}(R')$ and $r_M(X \cap R) = r_{M'}(X' \cap R')$. Then, by (4.2.3), we obtain $d_{A'}^-(X') = d_A^-(X) \geq r_M(R) - r_M(X \cap R) = r_{M'}(R') - r_{M'}(X' \cap R')$, that is (D', M') satisfies (4.1.2). We can now apply Theorem 4.1.7 to obtain in (D', M') a matroid-based arborescence packing $\{\mathbf{B}'_{r'}\}_{r' \in R'}$.

For all $r \in R$, let \mathbf{B}_r be obtained from $\{\mathbf{B}'_{r'}\}_{r' \in Q_r}$ by contracting all vertices of Q_r into r . Since $\{\mathbf{B}'_{r'}\}_{r' \in R'}$ is a packing, so is $\{\mathbf{B}_r\}_{r \in R}$. Let $r \in R$. Since $\{r' \in R' : v \in V(B'_{r'})\}$ is independent in M' for all $v \in V$ and Q_r is a set of parallel elements in M' , $\{\mathbf{B}'_{r'}\}_{r' \in Q_r}$ is a set of vertex-disjoint r' -arborescences in D' and hence B_r is an r -arborescence in D . Moreover, for all $v \in V$, we have that $\{r \in R : v \in V(B_r)\}$ is a basis of M because $\{r' \in R' : v \in V(B'_{r'})\}$ is a basis of M' . Thus, the packing $\{\mathbf{B}_r\}_{r \in R}$ of arborescences has the desired properties. \square

We are now ready to proceed to the main proof of Theorem 4.2.2.

Proof. (of **Theorem 4.2.2**) For necessity, suppose that (D, M) has a matroid-reachability-based packing $\{\mathbf{B}_r\}_{r \in R}$ and consider some $X \subseteq V \cup R$ with $X \cap V \neq \emptyset$. For every $v \in X$, let $\mathbf{R}_v = \{r \in R : v \in V(B_r)\}$ and let $\mathbf{R}_X = \cup_{v \in X} \mathbf{R}_v$. By Proposition 2.4.3 and as $\{\mathbf{B}_r\}_{r \in R}$ is a matroid-reachability-based packing, we obtain

$$\begin{aligned} \text{span}_M(R_X) &= \text{span}_M(\cup_{v \in X} R_v) \\ &\supseteq \cup_{v \in X} \text{span}_M(R_v) \\ &= \cup_{v \in X} (P_v^D \cap R) \\ &= P_X^D \cap R, \end{aligned}$$

so $r_M(R_X) \geq r_M(P_X^D \cap R)$. Now Proposition 2.4.1 yields

$$\begin{aligned} |R_X - X| &\geq r_M(R_X - X) \\ &\geq r_M(R_X) - r_M(X \cap R) \\ &\geq r_M(P_X^D \cap R) - r_M(X \cap R). \end{aligned}$$

As B_r contains an arc entering X for all $r \in R_X - X$ and the B_r are disjoint, the statement follows.

For sufficiency, let $(\mathbf{D} = (V \cup R, A), \mathbf{M} = (R, r_M))$ be a minimum counterexample. Obviously $V \neq \emptyset$. By Proposition 2.1.1 and as V has no arc leaving, there is a strongly connected component of $D[V]$ which has no arc leaving in D . We denote its vertex set by C .

Let $\mathbf{D}_1 = (\mathbf{V}_1 \cup R, \mathbf{A}_1) = D - C$. Note that (D_1, M) is a matroid-rooted digraph.

Lemma 4.2.4. (D_1, M) contains a matroid-reachability-based arborescence packing $\{B_r^1\}_{r \in R}$ and $P_v^{D_1} = P_v^D$ for all $v \in V_1$.

Proof. By $d_A^+(C) = 0$, we have $d_{A_1}^-(X) = d_A^-(X)$ and $P_X^{D_1} = P_X^D$ for all $X \subseteq V_1 \cup R$. Then, since D satisfies (4.2.2), so does D_1 . Hence, by the minimality of D and $P_v^{D_1} = P_v^D$ for all $v \in V_1$, the desired packing exists in D_1 . \square

By Lemma 4.2.4, (D_1, M) has a matroid-reachability-based arborescence packing $\{B_r^1\}_{r \in R}$. We now define an auxiliary matroid-rooted digraph $(\mathbf{D}_2, \mathbf{M}_2)$ which depends on these arborescences. Let $R_2 = P_C^D \cap R$ and \mathbf{M}_2 the restriction of M to R_2 . We now define $\mathbf{D}_2 = (V_2 \cup R_2, A_2)$. First let $\mathbf{V}_2 = C \cup T$, where $T = \{t_{uv} : uv \in \delta_A^-(C)\}$ is a set of new vertices. We first let \mathbf{A}_2 contain $A(D[C]) \cup \{rt_{uv} : r \in R_2, u \in V(B_r^1), t_{uv} \in T\}$. Further, for every $t_{uv} \in T$, we let A_2 contain one arc from t_{uv} to v and $r_{M_2}(R_2)$ arcs from v to t_{uv} .

Lemma 4.2.5. (D_2, M_2) has a matroid-based arborescence packing $\{B_r^2\}_{r \in R_2}$.

Proof. We show in the following claim that (D_2, M_2) satisfies (4.2.3). Let $X \subseteq V_2 \cup R_2$ with $X \cap V_2 \neq \emptyset$ and $X \cap R_2 = \text{span}_{M_2}(N_{D_2}^-(X \cap V_2) \cap R_2)$.

Claim 4.2.2. $d_{A_2}^-(X) \geq r_{M_2}(R_2) - r_{M_2}(X \cap R_2)$.

Proof. If $X \cap C = \emptyset$, we obtain from $X - R_2 \neq \emptyset$ that there is some $t_{uv} \in X$. By construction, this yields $d_{A_2}^-(X) \geq d_{A_2}(v, t_{uv}) = r_{M_2}(R_2) \geq r_{M_2}(R_2) - r_{M_2}(X \cap R_2)$.

If $X \cap C \neq \emptyset$, then, since $D[C]$ is strongly connected, we have $R_2 = P_C^D \cap R = P_{X \cap C}^D \cap R$. Let $\mathbf{Y} = (V \cup R) - U_{R-X}^D$ and $\mathbf{Z} = (X \cap C) \cup Y$. Observe that $P_Z^D \cap R = R_2$ and $Z \cap R = X \cap R_2$.

Proposition 4.2.1. $d_{A_2}^-(X) \geq d_A^-(Z)$.

Proof. Let $uv \in \delta_A^-(Z)$. Since, by the definition of Y , we have $\delta_A^-(Y) = \emptyset$, we obtain $v \in X \cap C$. If $u \in C$, then $uv \in \delta_{A_2}^-(X)$.

If $u \notin C$, then $u \in U_{R-X}^D$, so there is some $\bar{r} \in R - X$ such that $u \in U_{\bar{r}}^D$.

Further, as $uv \in \delta_A^-(C)$, we know that T contains the vertex t_{uv} and A_2 contains the arc $t_{uv}v$. Then, by $uv \in A$ and $v \in X \cap C$, we have $\bar{r} \in P_u^D \cap R \subseteq P_{X \cap C}^D \cap R = R_2$. Note that $\{r \in R : \bar{r} \in V(B_r^1)\} = \{\bar{r}\} = P_{\bar{r}}^D$. If $t_{uv} \in X$, then, since $\{r \in R : u \in V(B_r^1)\}$ is a basis of $P_u^D \cap R$ in M , we have

$$\begin{aligned} \bar{r} \notin X \cap R_2 &= \text{span}_{M_2}(N_{D_2}^-(X - R_2) \cap R_2) \\ &\supseteq \text{span}_{M_2}(N_{D_2}^-(t_{uv}) \cap R_2) \\ &= \text{span}_{M_2}(\{r \in R_2 : u \in V(B_r^1)\}) \\ &= \text{span}_M(\{r \in R : u \in V(B_r^1)\}) \cap R_2 \\ &\supseteq P_u^D \cap R_2 \supseteq \{\bar{r}\}, \end{aligned}$$

a contradiction. Thus $t_{uv} \notin X$ and so $t_{uv}v \in \delta_{A_2}^-(X)$. We have thus found an injection from the arcs entering Z in D to the arcs entering X in D_2 . \square

By Proposition 4.2.1 and (4.2.2), we have $d_{A_2}^-(X) \geq d_A^-(Z) \geq r_M(P_Z^D \cap R) - r_M(Z \cap R) = r_{M_2}(R_2) - r_{M_2}(X \cap R_2)$ and the proof of Claim 4.2.2 is complete. \square

By Claim 4.2.2 and Theorem 4.2.3, the desired packing exists in D_2 . This completes the proof of Lemma 4.2.5. \square

By Lemma 4.2.5, (D_2, M_2) has a matroid-based arborescence packing $\{\mathbf{B}_r^2\}_{r \in R_2}$. With the help of the packings $\{B_r^1\}_{r \in R}$ and $\{B_r^2\}_{r \in R_2}$, a packing in (D, M) can be constructed yielding a contradiction.

Lemma 4.2.6. *(D, M) has a matroid-reachability-based arborescence packing.*

Proof. For all $r \in R - R_2$, let $\mathbf{B}_r = B_r^1$ and for all $r \in R_2$, let \mathbf{B}_r be obtained from the union of B_r^1 and $B_r^2 - (R_2 \cup T)$ by adding the arc uv for all $t_{uv}v \in A(B_r^2)$. Since $\{B_r^1\}_{r \in R}$ and $\{B_r^2\}_{r \in R_2}$ are packings, so is $\{B_r\}_{r \in R}$. Since B_r^1 and B_r^2 are arborescences, for all $r \in R$ and $v \in V$, we have $d_{A(B_r)}^-(v) \leq 1$. Further, if $d_{A(B_r)}^+(v) \geq 1$, then $d_{A(B_r)}^-(v) = 1$ or $v = r$. It follows that B_r is an r -arborescence indeed. For $v \in V - C$, we have $\{r \in R : v \in V(B_r)\} = \{r \in R : v \in V(B_r^1)\}$ which is a basis of $P_v^D \cap R$ in M by Lemma 4.2.4. For $v \in C$, we have $\{r \in R : v \in V(B_r)\} = \{r \in R_2 : v \in V(B_r^2)\}$ which is a basis of M_2 , so a basis of $R_2 = P_v^D \cap R$ in M . It follows that $\{B_r\}_{r \in R}$ has indeed the desired properties. \square

Lemma 4.2.6 contradicts the fact that (D, M) is a counterexample and hence completes the proof of Theorem 4.2.2. \square

4.2.3 Reachability in packings of mixed hyperarborescences

Several extensions of Theorem 4.1.2 and Theorem 4.1.7 from digraphs to more general objects like mixed graphs and dypergraphs have been provided. Given a matroid-rooted mixed hypergraph $(F = (V \cup R, \mathcal{A} \cup \mathcal{E}), M)$, we say that a packing of mixed r -hyperarborescences $\{\mathcal{B}_r\}_{r \in R}$ is *matroid-reachability-based* if for every $r \in R$, there is a trimming B_r of \mathcal{B}_r such that $\{B_r : r \in R\}$ is matroid-reachability-based packing of arborescences. Observe that the reachability condition refers to the reachability in the original mixed hypergraph. A characterization for rooted dypergraphs admitting a reachability-based packing of mixed dyperarborescences was given by Bérczi and Frank [10]. This was generalized to matroid-reachability-based packings by Fortier et al. in [38].

The characterizations for the problems combining mixed graphs and reachability are somewhat more complicated as they include biset partitions. The case of reachability-based packings in mixed graphs was settled by Matsuoka and Tanigawa [92] and the case of matroid-reachability-based packings in mixed graphs was settled by Gao and Yang [54]. A careful analysis of the proof of Gao and Yang in [54] shows that it can be literally generalized to matroid-reachability-based packings in mixed hypergraphs. We use a different approach and settle this case by concluding it from Theorem 4.1.8. More concretely, we prove the following result.

Theorem 4.2.4. *Let $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}), M = (R, r_M))$ be a matroid-rooted mixed hypergraph. Then there exists a matroid-reachability-based mixed hyperarborescence packing in (\mathcal{F}, M) if and only if for every strongly connected component C of $\mathcal{F} - R$ and for every biset subpartition $\{X^i\}_1^\ell$ of C such that $w(X^i) = P_{w(X^i)}^{\mathcal{F}}$ for all $i = 1, \dots, \ell$, we have*

$$e_{\mathcal{E}}(\{X_I^i\}_1^\ell) + \sum_{i=1}^{\ell} d_{\mathcal{A}}^-(X^i) \geq \sum_{i=1}^{\ell} (r_M(P_C^{\mathcal{F}} \cap R) - r_M(X_O^i \cap R)). \quad (4.2.4)$$

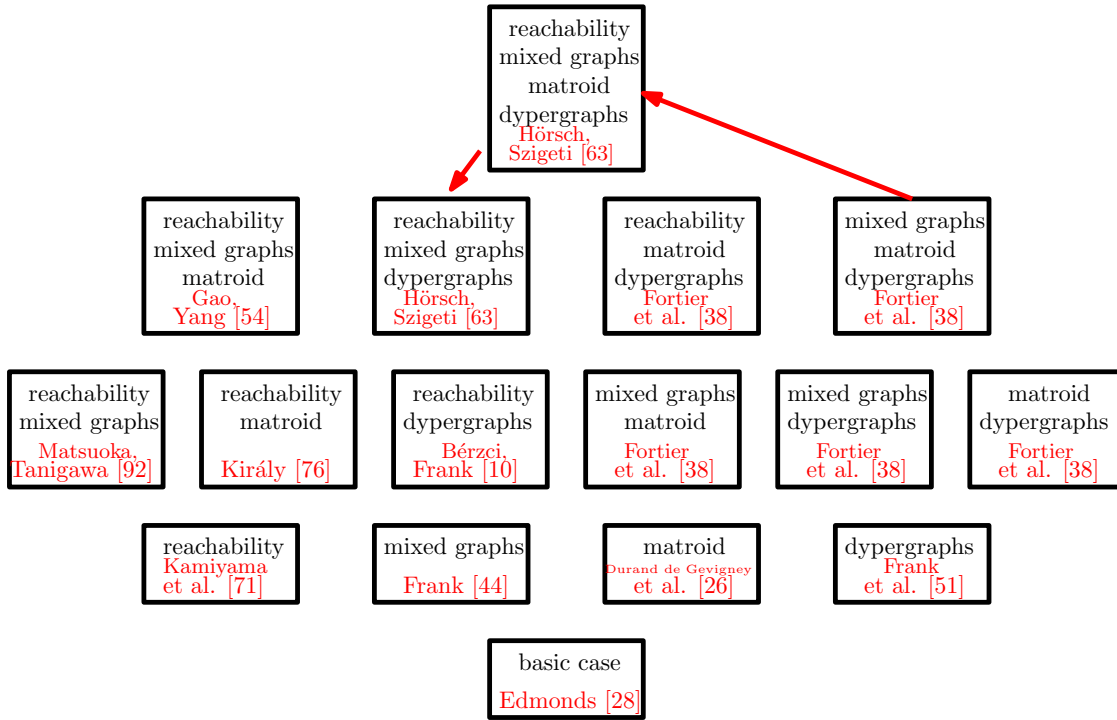


Figure 4.2.7: An illustration of the extensions of Theorem 4.1.2.

As a corollary, we obtain the following result which characterizes matroid-rooted mixed hypergraphs admitting a reachability-based packing of mixed hyperarborescences by applying Theorem 4.2.4 to the free matroid.

Corollary 4.2.1. *Let $\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E})$ be a rooted mixed hypergraph. Then there exists a reachability mixed hyperarborecence packing $\{\mathcal{B}_r\}_{r \in R}$ in \mathcal{F} if and only if for every strongly connected component C of $\mathcal{F} - R$ and for every biset subpartition $\{\mathbf{X}^i\}_1^\ell$ of C such that $w(\mathbf{X}^i) = P_{w(\mathbf{X}^i)}^\mathcal{F}$ for all $i = 1, \dots, \ell$, we have*

$$e_\mathcal{E}(\{\mathbf{X}_I^i\}_1^\ell) + \sum_{i=1}^{\ell} d_{\mathcal{A}}^-(\mathbf{X}^i) \geq \sum_{i=1}^{\ell} (|P_C^\mathcal{F} \cap R| - |\mathbf{X}_O^i \cap R|). \quad (4.2.5)$$

For an illustration of the extensions of Theorem 4.1.2 discussed in this chapter, see Figure 4.2.7.

Our proof of Theorem 4.2.4 uses Theorem 4.1.8 and follows the same ideas as the proofs of Theorem 4.2.1 and Theorem 4.2.2. Again, we first need a slightly stronger version of Theorem 4.1.8 where the simplicity condition is omitted.

Theorem 4.2.5. *Let $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}), M = (R, r_M))$ be a matroid-rooted mixed hypergraph. Then there exists a matroid-based mixed hyperarborecence packing in (\mathcal{F}, M) if and only if for every biset subpartition $\{\mathbf{X}^i\}_1^\ell$ of V with $w(\mathbf{X}^i) = \text{span}_M(\{r \in R : N_{\mathcal{F}}^+(r) \cap X_I^i \neq \emptyset\})$ for $i = 1, \dots, \ell$, we have*

$$e_\mathcal{E}(\{\mathbf{X}_I^i\}_1^\ell) + \sum_{i=1}^{\ell} d_{\mathcal{A}}^-(\mathbf{X}^i) \geq \sum_{i=1}^{\ell} (r_M(R) - r_M(w(\mathbf{X}^i))). \quad (4.2.6)$$

Proof. We first prove necessity. Suppose that there exists a matroid-based mixed hyperarborecence packing $\{\mathcal{B}_r\}_{r \in R}$. By definition, for every $r \in R$, there is an r -arborecence B_r that is a trimming of \mathcal{B}_r with $\{r \in R : v \in V(B_r)\}$ being a basis of M for all $v \in V$. Let $\{\mathbf{X}^i\}_1^\ell$ be a biset subpartition of V such that $w(\mathbf{X}^i) = \text{span}_M(\{r \in R : N_{\mathcal{F}}^+(r) \cap X_I^i \neq \emptyset\})$ for all $i = 1, \dots, \ell$.

Let $\mathbf{i} \in \{1, \dots, \ell\}$, $\mathbf{R}_i = \{r \in R - X_O^i : V(B_r) \cap X_I^i \neq \emptyset\}$ and $\mathbf{v} \in X_I^i$. By Proposition 2.4.1, we have

$$\begin{aligned} |R_i| &\geq r_M(R_i) \\ &\geq r_M(R_i \cup (X_O^i \cap R)) - r_M(X_O^i \cap R) \\ &\geq r_M(\{r \in R : \mathbf{v} \in V(B_r)\}) - r_M(X_O^i \cap R) \\ &= r_M(R) - r_M(X_O^i \cap R). \end{aligned}$$

Since $w(\mathbf{X}^i) \subseteq R$, no dyperedge and no hyperedge enters $w(\mathbf{X}^i)$ in \mathcal{F} . Then, as $w(\mathbf{X}^i) \cap R_i = \emptyset$, every B_r with $r \in R_i$ has an arc that enters \mathbf{X}^i , that is \mathcal{B}_r contains either a dyperedge in \mathcal{A} entering \mathbf{X}^i or a hyperedge in \mathcal{E} entering X_I^i . Thus, since $\{\mathcal{B}_r\}_{r \in R}$ is a packing and the X_I^i are disjoint, we have

$$\begin{aligned} e_{\mathcal{E}}(\{X_I^i\}_1^\ell) + \sum_{i=1}^{\ell} d_{\mathcal{A}}^-(\mathbf{X}^i) &\geq \sum_{i=1}^{\ell} |R_i| \\ &\geq \sum_{i=1}^{\ell} (r_M(R) - r_M(X_O^i \cap R)) \\ &= \sum_{i=1}^{\ell} (r_M(R) - r_M(w(\mathbf{X}^i))). \end{aligned}$$

For sufficiency, we define a simply matroid-rooted mixed hypergraph $(\mathcal{F}' = (\mathbf{V} \cup \mathbf{R}', \mathcal{A}' \cup \mathcal{E}), \mathbf{M}' = (\mathbf{R}', \mathbf{r}_{M'}))$ obtained from (\mathcal{F}, M) by replacing every root $r \in R$ by a set Q_r of $|N_{\mathcal{F}}^+(r)|$ simple roots such that $N_{\mathcal{F}'}^+(Q_r) = N_{\mathcal{F}}^+(r)$ in the mixed hypergraph and by $|Q_r|$ parallel copies of r in the matroid.

Now let $\{\mathbf{X}^i\}_1^\ell$ be a biset subpartition of V with $w(\mathbf{X}^i) = \text{span}_{M'}(\{r' \in R' : N_{\mathcal{F}'}^+(r') \cap X_I^i \neq \emptyset\})$ for $i = 1, \dots, \ell$. Let $\mathbf{i} \in \{1, \dots, \ell\}$. Note that for all $r \in R$, either $Q_r \subseteq w(\mathbf{X}^i)$ or $Q_r \cap w(\mathbf{X}^i) = \emptyset$. Let $\mathbf{Y}^i = (X_I^i \cup \{r \in R : Q_r \subseteq w(\mathbf{X}^i)\}, X_I^i)$. Observe that $w(\mathbf{Y}^i) = \text{span}_M(\{r \in R : N_{\mathcal{F}}^+(r) \cap X_I^i \neq \emptyset\})$, $d_{\mathcal{A}'}^-(\mathbf{Y}^i) = d_{\mathcal{A}}^-(\mathbf{X}^i)$, $r_M(R) = r_{M'}(R')$ and $r_M(w(\mathbf{Y}^i)) = r_{M'}(w(\mathbf{X}^i))$. Then, as $X_I^i = Z_I^i$ for $i = 1, \dots, \ell$ and by (4.2.6), we obtain

$$\begin{aligned} e_{\mathcal{E}}(\{X_I^i\}_1^\ell) &= e_{\mathcal{E}}(\{Y_I^i\}_1^\ell) \\ &\geq \sum_{i=1}^{\ell} (r_M(R) - r_M(w(\mathbf{Y}^i)) - d_{\mathcal{A}}^-(\mathbf{Y}^i)) \\ &= \sum_{i=1}^{\ell} (r_{M'}(R') - r_{M'}(w(\mathbf{X}^i)) - d_{\mathcal{A}'}^-(\mathbf{X}^i)), \end{aligned}$$

that is (\mathcal{F}', M') satisfies (4.1.3).

We now apply Theorem 4.1.8 to obtain in (\mathcal{F}', M') a matroid-based mixed hyperarborescences packing $\{\mathcal{B}'_{r'}\}_{r' \in R'}$ with arborescences $\{\mathcal{B}'_{r'}\}_{r' \in R'}$ as trimmings. For all $r \in R$, let \mathcal{B}_r be obtained from $\{\mathcal{B}'_{r'}\}_{r' \in Q_r}$ by contracting all vertices of Q_r into r . As in the proof of Theorem 4.2.3, we can see that $\{B_r\}_{r \in R}$ is a matroid-based arborescence packing. Finally, for all $r \in R$, let \mathcal{B}_r be obtained from $\{\mathcal{B}'_{r'}\}_{r' \in Q_r}$ by contracting all vertices of Q_r into r . As B_r is a trimming of \mathcal{B}_r for all $r \in R$, $\{\mathcal{B}_r\}_{r \in R}$ is a packing of mixed hyperarborescences with the desired properties. \square

We are now ready to proceed to the main proof of Theorem 4.2.4.

Proof. (of **Theorem 4.2.4**)

We first prove necessity. Suppose that there exists a matroid-reachability-based mixed hyperarborescence packing $\{\mathcal{B}_r\}_{r \in R}$. By definition, for every $r \in R$, there is an r -arborescence B_r that is a trimming of \mathcal{B}_r with $\{r \in R : v \in V(B_r)\}$ being a basis of $P_v^{\mathcal{F}} \cap R$ in M for all $v \in V$. Let $\{\mathbf{X}^i\}_1^\ell$ be a biset subpartition of a strongly connected component C of $\mathcal{F} - R$ such that $w(\mathbf{X}^i) = P_{w(\mathbf{X}^i)}^{\mathcal{F}}$ for all $i = 1, \dots, \ell$.

Let $\mathbf{i} \in \{1, \dots, \ell\}$, $\mathbf{R}_i = \{r \in R - X_O^i : V(B_r) \cap X_I^i \neq \emptyset\}$ and $\mathbf{v} \in X_I^i$. Then we have

$$r_M(\mathbf{R}_i \cup (X_O^i \cap R)) \geq r_M(\{r \in R : v \in V(B_r)\}) = r_M(P_v^{\mathcal{F}} \cap R) = r_M(P_C^{\mathcal{F}} \cap R).$$

Thus, by Proposition 2.4.1, we have

$$|\mathbf{R}_i| \geq r_M(\mathbf{R}_i) \geq r_M(\mathbf{R}_i \cup (X_O^i \cap R)) - r_M(X_O^i \cap R) \geq r_M(P_C^{\mathcal{F}} \cap R) - r_M(X_O^i \cap R).$$

Since $w(\mathbf{X}^i) = P_{w(\mathbf{X}^i)}^{\mathcal{F}}$, no dyperedge and no hyperedge enters $w(\mathbf{X}^i)$ in \mathcal{F} . Then, by $w(\mathbf{X}^i) \cap R_i = \emptyset$, every B_r with $r \in R_i$ has an arc that enters \mathbf{X}^i , that is \mathcal{B}_r contains either a dyperedge in \mathcal{A} entering \mathbf{X}^i or a hyperedge in \mathcal{E} entering X_I^i . Thus, since $\{\mathcal{B}_r\}_{r \in R}$ is a packing and the X_I^i are disjoint, we have

$$e_{\mathcal{E}}(\{X_I^i\}_1^\ell) + \sum_{i=1}^{\ell} d_{\mathcal{A}}^-(\mathbf{X}^i) \geq \sum_{i=1}^{\ell} |\mathbf{R}_i| \geq \sum_{i=1}^{\ell} (r_M(P_C^{\mathcal{F}} \cap R) - r_M(X_O^i \cap R)).$$

For sufficiency, let $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}), \mathbf{M} = (R, r_M))$ be a minimum counterexample. Obviously, $V \neq \emptyset$. By Proposition 2.1.5 and as there is no dyperedge or hyperedge leaving V in \mathcal{F} , there is a strongly connected component of $\mathcal{F}[V]$ which has no dyperedge or hyperedge leaving. We denote its vertex set by C .

Let $\mathcal{F}_1 = (V_1 \cup R, \mathcal{A}_1 \cup \mathcal{E}_1) = \mathcal{F} - C$. Note that (\mathcal{F}_1, M) is a matroid-rooted mixed hypergraph.

Lemma 4.2.7. (\mathcal{F}_1, M) has a matroid-reachability-based mixed hyperarborescence packing $\{\mathcal{B}_r^1\}_{r \in R}$ and $P_v^{\mathcal{F}_1} = P_v^{\mathcal{F}}$ for all $v \in V_1$.

Proof. The fact that $d_{\mathcal{A}}^+(C) = d_{\mathcal{E}}(C) = 0$ implies that for all $X \subseteq V_1 \cup R$, we have $P_X^{\mathcal{F}_1} = P_X^{\mathcal{F}}$, for every subpartition \mathcal{P} of $V \cup R_1$, we have $e_{\mathcal{E}}(\mathcal{P}) = e_{\mathcal{E}_1}(\mathcal{P})$, and for every biset \mathbf{X} on $V_1 \cup R$, we have $d_{\mathcal{A}_1}^-(\mathbf{X}) = d_{\mathcal{A}}^-(\mathbf{X})$. Then, since (\mathcal{F}, M) satisfies (4.2.4), so does (\mathcal{F}_1, M) . Hence, by the minimality of \mathcal{F} and $P_v^{\mathcal{F}_1} = P_v^{\mathcal{F}}$ for all $v \in V_1$, the desired packing exists. \square

By Lemma 4.2.7, (\mathcal{F}_1, M) has a matroid-reachability-based mixed hyperarborescence packing $\{\mathcal{B}_r^1\}_{r \in R}$. By definition, \mathcal{B}_r^1 can be trimmed to an r -arborescence B_r^1 for all $r \in R$ such that $\{r \in R : v \in V(B_r^1)\}$ is a basis of $P_v^{\mathcal{F}_1} \cap R = P_v^{\mathcal{F}} \cap R$ in M for all $v \in V_1$.

We now define a matroid-rooted mixed hypergraph $(\mathcal{F}_2 = (V_2 \cup R_2, \mathcal{A}_2 \cup \mathcal{E}_2), M_2)$ which depends on the arborescences $\{B_r^1\}_{r \in R}$. Let $\mathbf{R}_2 = P_C^{\mathcal{F}} \cap R$ and \mathbf{M}_2 the restriction of M to R_2 . Next, let \mathbf{V}_2 contain C and a set \mathbf{T} which contains a vertex t_a for every $a \in \delta_{\mathcal{A}}^-(C)$. Firstly, \mathcal{A}_2 contains all dyperedges whose head and tail are completely contained in C . Next for all $a \in \delta_{\mathcal{A}}^-(C)$, \mathcal{A}_2 contains a dyperedge a' whose tail is $(tail(a) \cap C) \cup t_a$ and whose head is $head(a)$ and $r_{M_2}(R_2)$ parallel arcs $head(a)t_a$. Further, \mathcal{A}_2 contains the arc rt_a for $r \in R_2$ and $a \in \delta_{\mathcal{A}}^-(C)$ whenever $tail(a) \cap V(B_r^1) \neq \emptyset$. Finally, we let \mathcal{E}_2 contain all the hyperedges which are completely contained in C . Observe that \mathcal{E}_2 contains no hyperedge entering C .

Lemma 4.2.8. (\mathcal{F}_2, M_2) contains a matroid-based mixed hyperarborescence packing $\{\mathcal{B}_r^2\}_{r \in R_2}$.

Proof. We show in the following claim that (\mathcal{F}_2, M_2) satisfies (4.2.6). Let $\{\mathbf{X}^i\}_1^\ell$ be a biset subpartition of $V_2 = C \cup T$ with $w(\mathbf{X}^i) = \text{span}_{M_2}(\{r \in R_2 : N_{\mathcal{F}_2}^+(r) \cap X_I^i \neq \emptyset\})$ for all $i = 1, \dots, \ell$.

Claim 4.2.3. $e_{\mathcal{E}_2}(\{X_I^i\}_1^\ell) \geq \sum_{i=1}^\ell (r_{M_2}(R_2) - r_{M_2}(w(\mathbf{X}^i)) - d_{\mathcal{A}_2}^-(\mathbf{X}^i)).$

Proof. Suppose that $X_I^i \cap C \neq \emptyset$ for all $i \in \{1, \dots, j\}$ and $X_I^i \cap C = \emptyset$ for all $i \in \{j+1, \dots, \ell\}$. For $i \in \{j+1, \dots, \ell\}$, as $X_I^i - V_2 \neq \emptyset$, there is some $a \in \delta_{\mathcal{A}}^-(C)$ such that $t_a \in X_I^i$. As $w(\mathbf{X}^i) \subseteq R$, it follows that $\text{head}(a) \in V_2 - X_O^i$. This yields $d_{\mathcal{A}_2}^-(\mathbf{X}^i) \geq d_{\mathcal{A}_2}(\text{head}(a), t_a) \geq r_{M_2}(R_2)$. We obtain $0 \geq r_{M_2}(R_2) - r_{M_2}(w(\mathbf{X}^i)) - d_{\mathcal{A}_2}^-(\mathbf{X}^i)$.

Let now $i \in \{1, \dots, j\}$. Since $\mathcal{F}[C]$ is strongly connected, we have $R_2 = P_C^{\mathcal{F}} \cap R = P_{X_I^i \cap C}^{\mathcal{F}} \cap R$. Let $\mathbf{Y}^i = (V \cup R) - (U_{R-w(\mathbf{X}^i)}^{\mathcal{F}} \cup C)$ and $\mathbf{Z}^i = ((X_I^i \cap C) \cup Y^i, X_I^i \cap C)$. Note that $Z_I^i = X_I^i \cap C$ and $Z_O^i \cap R = Y^i \cap R = R - (R - w(\mathbf{X}^i)) = w(\mathbf{X}^i)$, so $r_M(Z_O^i \cap R) = r_{M_2}(w(\mathbf{X}^i))$.

Proposition 4.2.2. $d_{\mathcal{A}_2}^-(\mathbf{X}^i) \geq d_{\mathcal{A}}^-(\mathbf{Z}^i).$

Proof. Let $\mathbf{a} \in \delta_{\mathcal{A}}^-(\mathbf{Z}^i)$.

If $(\text{tail}(\mathbf{a}) - Z_O^i) \cap C \neq \emptyset$ and $(\text{tail}(\mathbf{a}) - C) \neq \emptyset$, then $\mathbf{a}' \in \delta_{\mathcal{A}_2}^-(\mathbf{X}^i)$. If $(\text{tail}(\mathbf{a}) - Z_O^i) \cap C \neq \emptyset$ and $(\text{tail}(\mathbf{a}) - C) = \emptyset$, then $\mathbf{a} \in \delta_{\mathcal{A}_2}^-(\mathbf{X}^i)$.

We may hence suppose that $(\text{tail}(\mathbf{a}) - Z_O^i) \cap C = \emptyset$. Let $\mathbf{u} \in \text{tail}(\mathbf{a}) - Z_O^i - C$, so $\mathbf{u} \in V \cup R - C - Y^i \subseteq U_{R-w(\mathbf{X}^i)}^{\mathcal{F}}$. It follows that $\mathbf{u} \in U_{\bar{r}}^{\mathcal{F}}$ for some $\bar{r} \in R - w(\mathbf{X}^i)$. Further, as $\mathbf{a} \in \delta_{\mathcal{A}}^-(C)$, we know that T contains the vertex t_a and \mathcal{A}_2 contains the arc \mathbf{a}' .

Thus, by $\mathbf{a} \in \mathcal{A}$, we have $\bar{r} \in P_{\mathbf{u}}^{\mathcal{F}} \cap R \subseteq P_{X_I^i \cap C}^{\mathcal{F}} \cap R = R_2$. Note that $\{r \in R : \bar{r} \in V(B_r^1)\} = \{\bar{r}\} = P_{\bar{r}}^{\mathcal{F}}$. If $t_a \in X_I^i$, then, since $\{r \in R : \mathbf{u} \in V(B_r^1)\}$ is a basis of $P_{\mathbf{u}}^{\mathcal{F}} \cap R$ in M , we obtain

$$\begin{aligned} \bar{r} \notin w(\mathbf{X}^i) &= \text{span}_{M_2}(\{r \in R_2 : N_{\mathcal{F}_2}^+(r) \cap X_I^i \neq \emptyset\}) \\ &\supseteq \text{span}_{M_2}(\{r \in R_2 : t_a \in N_{\mathcal{F}_2}^+(r)\}) \\ &= \text{span}_{M_2}(\{r \in R_2 : \text{tail}(\mathbf{a}) \cap V(B_r^1) \neq \emptyset\}) \\ &\supseteq \text{span}_M(\{r \in R : \mathbf{u} \in V(B_r^1)\}) \cap R_2 \\ &\supseteq P_{\mathbf{u}}^{\mathcal{F}} \cap R_2 \supseteq \{\bar{r}\}, \end{aligned}$$

a contradiction. It follows that $t_a \notin X_I^i$ and hence $t_a \notin X_O^i$. This yields that $\mathbf{a}' \in \delta_{\mathcal{A}_2}^-(\mathbf{X}^i)$. \square

Since $w(\mathbf{Z}^i) \cap C = \emptyset$, we have that $\{Z_I^i\}_1^j$ is a biset subpartition of C . Moreover, no dypredge and no hyperedge leaves $U_{R-w(\mathbf{X}^i)}^{\mathcal{F}} \cup C$, so $w(\mathbf{Z}^i) = Y^i = P_{Y^i}^{\mathcal{F}} = P_{w(\mathbf{Z}^i)}^{\mathcal{F}}$. Then, as $X_I^i \cap C = Z_I^i \cap C$ for $i = 1, \dots, j$, $\mathcal{E}_2 = \mathcal{E}[C]$, by (4.2.4) and Proposition 4.2.2, we have

$$\begin{aligned} e_{\mathcal{E}_2}(\{X_I^i\}_1^\ell) &= e_{\mathcal{E}_2}(\{X_I^i\}_1^j) \\ &= e_{\mathcal{E}}(\{Z_I^i\}_1^j) \\ &\geq \sum_{i=1}^j (r_M(P_C^{\mathcal{F}} \cap R) - r_M(Z_O^i \cap R) - d_{\mathcal{A}}^-(\mathbf{Z}^i)) \\ &\geq \sum_{i=1}^j (r_{M_2}(R_2) - r_{M_2}(w(\mathbf{X}^i)) - d_{\mathcal{A}_2}^-(\mathbf{X}^i)) \\ &\geq \sum_{i=1}^\ell (r_{M_2}(R_2) - r_{M_2}(w(\mathbf{X}^i)) - d_{\mathcal{A}_2}^-(\mathbf{X}^i)), \end{aligned}$$

that completes the proof of Claim 4.2.3. \square

By Claim 4.2.3 and Theorem 4.2.5, the desired packing exists in \mathcal{F}_2 . \square

By Lemma 4.2.8, (\mathcal{F}_2, M_2) has a matroid-reachability-based mixed hyperarborescence packing $\{\mathcal{B}_r^2\}_{r \in R_2}$ with r -arborescences $\{B_r^2\}_{r \in R_2}$ as trimmings. With the help of the packings $\{\mathcal{B}_r^1\}_{r \in R}$ and $\{\mathcal{B}_r^2\}_{r \in R_2}$, a packing of (\mathcal{F}, M) can be constructed yielding a contradiction.

Lemma 4.2.9. *(\mathcal{F}, M) has a matroid-reachability-based mixed hyperarborescence packing.*

Proof. For $r \in R - R_2$, let $\mathcal{B}_r = B_r^1$ and for $r \in R_2$, let \mathcal{B}_r be obtained from the union of B_r^1 and $B_r^2 - R_2 - T$ by adding an arc uv for all $t_a v \in \mathcal{A}(B_r^2)$ for some $u \in \text{tail}(a) \cap V(B_r^1)$. As in the proof of Theorem 4.2.2, we can see that $\{B_r\}_{r \in R}$ is a packing of arborescences such that the root of B_r is r for all $r \in R$ and $\{r \in R : v \in V(B_r)\}$ is a basis of $P_v^{\mathcal{F}} \cap R$ in M for all $v \in V$.

Finally, for $r \in R - R_2$, let $\mathcal{B}_r = B_r^1$ and for $r \in R_2$, let \mathcal{B}_r be obtained from B_r^1 and $B_r^2 - R_2 - T$ by adding the dyperedge $a \in \mathcal{A}$ for all $a' \in \mathcal{A}(B_r^2)$. The above argument shows that this is a packing of mixed hyperarborescences in \mathcal{F} (with arborescences $\{B_r\}_{r \in R}$ as trimmings) with the desired properties. \square

Lemma 4.2.9 contradicts the fact that (\mathcal{F}, M) is a counterexample and hence the proof of Theorem 4.2.4 is complete. \square

4.2.4 Algorithmic aspects

Again, there are two algorithmic problems we may want to consider. Firstly, we may want to find some arbitrary packing of reachability arborescences and secondly, we can search for such a packing of minimum weight.

We first deal with the unweighted case. Again, observe that we wish to find the actual arborescences, not just the arc set of the packing.

For the basic setting, we show that our proof of Theorem 4.2.1 yields a polynomial time algorithm for finding the desired packing of reachability arborescences. We acknowledge that so does the original proof in [71]. We first mention that the packings in Theorem 4.1.3 can be found in polynomial time, following the proof of Frank (Theorem 10.2.1 in [41]). Using this, we can turn our proof of Theorem 4.2.1 into a recursive polynomial time algorithm for finding the desired packing of arborescences. Recursively, we first find the arborescences B_r^1 in the smaller instance $D - C$ in polynomial time. As the size of D_2 is polynomial in the size of D , we can apply the algorithm mentioned above to obtain the arborescences B_r^2 in polynomial time. The obtained arborescences can be merged efficiently to obtain the arborescences B_r .

For the matroidal setting, we show that our proof of Theorem 4.2.2 yields an algorithm for finding the matroid-reachability-based arborescence packing if an independence oracle for M is given. We acknowledge that so does the original proof in [76]. We first recall that the packings in Theorem 4.1.7 can be found in polynomial time given an independence oracle for M as mentioned in [26]. It is easy to see that using this algorithm the proof of Theorem 4.2.3 yields a polynomial time algorithm for finding a matroid-based packing of arborescences in a matroid-rooted digraph if an independence oracle for M is given. By similar arguments as before and the fact that an independence oracle for M yields independence oracles for all matroids considered, we obtain that the proof of Theorem 4.2.2 can be turned into a polynomial time algorithm if an independence oracle for M is available.

For the most general setting, using the fact that the proof of Theorem 4.1.8 is algorithmic if an independence oracle is given ([38]), we obtain that also Theorems 4.2.5

and 4.2.4 yield polynomial time algorithms given independence oracles. In particular, the arborescences in Corollary 4.2.1 can be found in polynomial time.

We now deal with the weighted case. Observe that here it suffices to find the arc set of the minimum weight packing as we can then find the arborescences by applying the above algorithm to the rooted digraph which is restricted to these arcs. We first wish to mention that there is a result of Király, Szigeti and Tanigawa that shows that the arc sets of the matroid-reachability-based arborescence packings in matroid-singularly rooted graphs can be modeled as the intersection of two matroids [79]. We thus obtain a polynomial time algorithm for the problem of finding a matroid-reachability-based arborescence packing of minimum weight whenever an independence oracle for the input matroid is available. Their matroid construction is pretty involved. Nevertheless, this has recently been generalized to matroid-reachability-based packings of mixed hypergraphs by Király [78]. We follow a different approach. Based on our proof of Theorems 4.2.1, 4.2.2 and 4.2.4, we give an algorithm that only uses basic methods apart from applying the corresponding results for the problems where the reachability condition is omitted.

We give a more detailed description of this algorithm for the most general setting discussed in this section. More concretely, we show how to obtain the following result from Theorem 4.1.16 and the proof of Theorems 4.2.5 and 4.2.4.

Theorem 4.2.6. *Given a simply matroid-rooted mixed hypergraph $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}), M)$ with an independence oracle for M being available and a cost function $w : \mathcal{A} \cup \mathcal{E} \rightarrow \mathbb{R}$, a matroid-reachability-based packing of mixed hyperarborescences \mathcal{B} minimizing $w(\mathcal{B})$ can be computed in polynomial time if such a packing exists.*

In order to prove this, we first need the following slight strengthening of Theorem 4.1.16.

Theorem 4.2.7. *Given a matroid-rooted mixed hypergraph $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}), M)$ with an independence oracle for M being available and a cost function $w : \mathcal{A} \cup \mathcal{E} \rightarrow \mathbb{R}$, a matroid-based packing of mixed hyperarborescences \mathcal{B} minimizing $w(\mathcal{B})$ can be computed in polynomial time if such a packing exists.*

Proof. Let $(\mathcal{F} = (V \cup R, \mathcal{A} \cup \mathcal{E}), M)$ be a matroid-rooted mixed hypergraph. Let (\mathcal{F}', M') be obtained like in the proof of Theorem 4.2.5 and let $w' : \mathcal{E} \cup \mathcal{A}' \rightarrow \mathbb{R}$ be obtained by $w'(e) = w(e)$ for all $e \in \mathcal{E}$, $w'(a) = w(a)$ for all $a \in \mathcal{A} - \delta_{\mathcal{A}}^+(R')$ and $w'(a) = w(a')$ for all $a \in \delta_{\mathcal{A}}(Q_r, V)$ for all $r \in R'$ where a' is the arc from r to V that corresponds to a . It is easy to see that a matroid-based packing of mixed hyperarborescences in (\mathcal{F}, M) yields a matroid-based packing of mixed hyperarborescences in (\mathcal{F}', M') of the same weight and vice-versa. Hence a minimum weight solution for the problem in (\mathcal{F}', M') yields a minimum weight solution in (\mathcal{F}, M) . By Theorem 4.1.16, the minimum weight solution in (\mathcal{F}', M') can be computed in polynomial time. \square

We now turn our attention to the proof of Theorem 4.2.4. The following observation is crucial and an immediate consequence of the proof of Theorem 4.2.4.

Lemma 4.2.10. *Every matroid-reachability-based packing of $(\mathcal{F} - C, M)$ can be extended to a matroid-reachability-based packing of (\mathcal{F}, M) .*

This yields the following stronger statement.

Lemma 4.2.11. *Let $\mathcal{B}^1 = \{\mathcal{B}_r^1\}_{r \in R}$ be a matroid-reachability-based packing of $(\mathcal{F} - C, M)$ minimizing $w(\mathcal{B}^1)$. Then \mathcal{B}^1 can be extended to a matroid-reachability-based packing \mathcal{B} of (\mathcal{F}, M) minimizing $w(\mathcal{B})$.*

Proof. Let $\mathcal{B}^* = \{\mathcal{B}_r^*\}_{r \in R}$ be a matroid-reachability-based packing of (\mathcal{F}, M) minimizing $w(\mathcal{B}^*)$ and consider $\mathcal{F}' = (V \cup R, \mathcal{A}' \cup \mathcal{E}')$ with $\mathcal{A}' = \mathcal{A} - (\mathcal{A}(C) \cup \delta_{\mathcal{A}}^-(C) - \mathcal{A}(\mathcal{B}^*))$ and $\mathcal{E}' = \mathcal{E} - (\mathcal{E}(C) - \mathcal{E}(\mathcal{B}^*))$. Observe that \mathcal{B}^* is a matroid-reachability-based packing in (\mathcal{F}', M) and \mathcal{B}^1 is a matroid-reachability-based packing in $(\mathcal{F}' - C, M)$. By Lemma 4.2.10, \mathcal{B}^1 can be extended to a matroid-reachability-based packing in (\mathcal{F}', M) . Due to cardinality considerations, we obtain that $(\mathcal{A}(\mathcal{B}^1) \cup \mathcal{E}(\mathcal{B}^1)) \cup ((\mathcal{A}(\mathcal{B}^*) \cup \mathcal{E}(\mathcal{B}^*)) \cap (\mathcal{A}(C) \cup \delta_{\mathcal{A}}^-(C) \cup \mathcal{E}(C)))$ is the dyper- and hyperedge set of a matroid-reachability-based packing in (\mathcal{F}', M) and therefore also in (\mathcal{F}, M) . By the minimality of \mathcal{B}^1 , we obtain that \mathcal{B}^1 can be extended in the desired way. \square

We are now ready to proceed to the main proof of Theorem 4.2.6.

Proof. (of Theorem 4.2.6). By the above remarks, we may suppose that (\mathcal{F}, M) has a matroid-reachability-based packing of mixed hyperarborescences. Inductively, we may suppose that a matroid-reachability-based packing \mathcal{B}^1 in $(\mathcal{F} - C, M)$ minimizing $w(\mathcal{B}^1)$ can be constructed in polynomial time. Now construct the auxiliary matroid-rooted mixed hypergraph (\mathcal{F}_2, M_2) like in the proof of Theorem 4.2.4 and define the weight function $w': \mathcal{A}_2 \cup \mathcal{E}_2 \rightarrow \mathbb{R}$ by $w'(a) = w(a)$ for all $a \in \mathcal{A}(C)$, $w'(e) = w(e)$ for all $e \in \mathcal{E}(C)$, $w'(a') = w(a)$ for all $a' \in \delta_{\mathcal{A}_2}^-(C)$ and $w'(a) = 0$ all $a \in \delta_{\mathcal{A}_2}^-(T)$. By Lemma 4.2.11, \mathcal{B}^1 can be extended to a matroid-reachability-based packing of mixed hyperarborescences of (\mathcal{F}, M) and so (\mathcal{F}_2, M_2) has a matroid-based packing of mixed hyperarborescences. By Theorem 4.2.7, a matroid-based packing of mixed hyperarborescences \mathcal{B}^2 in (\mathcal{F}_2, M_2) minimizing $w'(\mathcal{B}^2)$ can be computed in polynomial time. We now obtain \mathcal{B} by merging \mathcal{B}^1 and \mathcal{B}^2 as in the proof of Theorem 4.2.4. It is easy to see that \mathcal{B} minimizes $w(\mathcal{B})$ among all the matroid-reachability-based packings extending \mathcal{B}^1 . By Lemma 4.2.10, $w(\mathcal{B})$ is minimal among all matroid-reachability-based packings of mixed hyperarborescences of (\mathcal{F}, M) . \square

4.3 Packings of mixed hyperarborescences with flexible roots

This section is based on [62]. We deal with a generalization of Theorem 4.1.2 in which the conditions of the arborescences being fixed is relaxed. Given a digraph, we wish to find a packing of a fixed number of arborescences without specifying their roots. However, we prescribe for every vertex of the digraph a lower and an upper bound on the number of arborescences whose root is this vertex.

4.3.1 Previous work

Given a digraph $D = (V, A)$, an integer $k \in \mathbb{Z}_{\geq 0}$ and functions $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$, we say that a packing of spanning arborescences is (k, f, g) -flexible if it contains k spanning arborescences and every $v \in V$ is the root of at least $f(v)$ and at most $g(v)$ of them. In the basic setting, the problem has been successfully treated by Frank. In [44], he proved the following theorem.

Theorem 4.3.1. *Let $D = (V, A)$ be a digraph, $k \in \mathbb{Z}_{\geq 0}$ and $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$ functions. There exists a (k, f, g) -flexible packing of spanning arborescences in \mathcal{F} if and only if we have*

$$f(v) \leq g(v) \quad \text{for every } v \in V, \quad (4.3.1)$$

$$e_A(\mathcal{P}) \geq k(|\mathcal{P}| - 1) + f(V - \cup \mathcal{P}) \quad \text{for every subpartition } \mathcal{P} \text{ of } V, \quad (4.3.2)$$

$$d_A^-(X) \geq k - g(X) \quad \text{for every nonempty } X \subseteq V. \quad (4.3.3)$$

Observe that if $f = g$, then Theorem 4.3.1 reduces to Theorem 4.1.2, so Theorem 4.3.1 generalizes Theorem 4.1.2 indeed.

Again, extensions of this result to mixed graphs and dypergraphs can be considered. A packing of mixed hyperarborescences $\{\mathcal{B}_i\}_{i \in \{1, \dots, k\}}$ is called (k, f, g) -flexible if there is a trimming B_i of \mathcal{B}_i for every $i = 1, \dots, k$ such that $\{B_i\}_{i \in \{1, \dots, k\}}$ is a (k, f, g) -flexible packing of arborescences. We first wish to mention that Theorem 4.3.1 can readily be generalized to dypergraphs using the reduction in [38]. The following extension to mixed graphs has recently been found by Gao and Yang [55].

Theorem 4.3.2. *Let $F = (V, A \cup E)$ be a mixed graph, $k \in \mathbb{Z}_{\geq 0}$ and $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$ functions. There exists a (k, f, g) -flexible packing of mixed arborescences in \mathcal{F} if and only if we have*

$$f(v) \leq g(v) \quad \text{for every } v \in V, \quad (4.3.4)$$

$$e_{E \cup A}(\mathcal{P}) \geq k(|\mathcal{P}| - 1) + f(V - \cup \mathcal{P}) \quad \text{for every subpartition } \mathcal{P} \text{ of } V, \quad (4.3.5)$$

$$e_{E \cup A}(\mathcal{P}) \geq k|\mathcal{P}| - g(\cup \mathcal{P}) \quad \text{for every subpartition } \mathcal{P} \text{ of } V. \quad (4.3.6)$$

We wish to remark that the proof of Theorem 4.3.2 does not easily generalize to mixed hypergraphs. We give an extension to mixed hypergraphs by proving the following result.

Theorem 4.3.3. *Let $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ be a mixed hypergraph, $k \in \mathbb{Z}_{\geq 0}$ and $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$ functions. There exists a (k, f, g) -flexible packing of mixed hyperarborescences in \mathcal{F} if and only if we have*

$$f(v) \leq g(v) \quad \text{for every } v \in V, \quad (4.3.7)$$

$$e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}) \geq k(|\mathcal{P}| - 1) + f(V - \cup \mathcal{P}) \quad \text{for every subpartition } \mathcal{P} \text{ of } V, \quad (4.3.8)$$

$$e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}) \geq k|\mathcal{P}| - g(\cup \mathcal{P}) \quad \text{for every subpartition } \mathcal{P} \text{ of } V. \quad (4.3.9)$$

Even though Theorem 4.3.3 is a literal generalization of Theorem 4.3.2, the approach of our proof is completely different from the one used in [55]. Our proof is based on matroid intersection. To our best knowledge, this is the first time in the context of arborescence packing that matroid intersection is used not only for algorithmic consequences but also in order to obtain a characterization. We also obtain the following algorithmic result.

Theorem 4.3.4. *Let $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ be a mixed hypergraph, $k \in \mathbb{Z}_{\geq 0}$, $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$ functions and $w : \mathcal{A} \cup \mathcal{E} \rightarrow \mathbb{R}$ a weight function. Then a (k, f, g) -flexible packing of mixed hyperarborescences of minimum weight can be computed in polynomial time, if there exists one.*

We wish to mention that there is no natural combination of matroid-based arborescence packings and arborescence packings with flexible roots. A combination of flexible roots and reachability is more interesting. However, a more restricted form of this problem has been proven to be NP-complete by Bérczi and Frank in [10].

4.3.2 Matroid intersection

We here describe how to model the problem of packing spanning mixed hyperarborescences with flexible roots via matroid intersection. The basic insight of our approach is that, given a packing of k spanning arborescences, for every vertex $v \in V$, the number of arborescences in the packing whose root is v plus the in-degree of v in the packing is equal to k . Given a mixed hypergraph $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$, we say that a packing of k mixed spanning dyperarborescences in its directed extension $\mathcal{D}_{\mathcal{F}}$ is (k, f, g) -feasible if every $v \in V$ is the root of at least $f(v)$ and at most $g(v)$ of the hyperarborescences and the packing contains at most one dyperedge of \mathcal{A}_e for all $e \in \mathcal{E}$. The following simple result relates (k, f, g) -feasible packings and (k, f, g) -flexible packings.

Lemma 4.3.1. *Let $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ be a mixed hypergraph, $\mathcal{D}_{\mathcal{F}} = (V, \mathcal{A} \cup \mathcal{A}_{\mathcal{E}})$ its directed extension, $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$ integer valued functions and $k \in \mathbb{Z}_{\geq 0}$. Then \mathcal{F} has a (k, f, g) -flexible packing if and only if $\mathcal{D}_{\mathcal{F}}$ has a (k, f, g) -feasible packing.*

Proof. First suppose that \mathcal{F} has a (k, f, g) -flexible packing $\mathcal{B} = \{\mathcal{B}_i : i \in \{1, \dots, k\}\}$. Then there is a (k, f, g) -flexible packing $\mathcal{B} = \{\mathcal{B}_i : i \in \{1, \dots, k\}\}$ of arborescences such that B_i is a trimming of \mathcal{B}_i for all $i \in \{1, \dots, k\}$. For every $e \in \mathcal{E}(\mathcal{B})$, let v_e be the head of the arc to which e is trimmed in B and let $\vec{e} \in \mathcal{A}_e$ be the orientation of e where v_e is chosen to be its head. Then the set of dyperedges $\mathcal{A}(\mathcal{B}) \cup \{\vec{e} : e \in \mathcal{E}(\mathcal{B})\}$ can be trimmed to B and it contains at most one dyperedge of \mathcal{A}_e for all $e \in \mathcal{E}$. It follows that $\mathcal{D}_{\mathcal{F}}$ has a (k, f, g) -feasible packing.

Now suppose that $\mathcal{D}_{\mathcal{F}}$ has a (k, f, g) -feasible packing \mathcal{B} . By definition, $\mathcal{A}_{\mathcal{E}} \cap \mathcal{A}(\mathcal{B})$ contains at most one dyperedge in \mathcal{A}_e for all $e \in \mathcal{E}$. Replacing a dyperedge in $\mathcal{A}_e \cap \mathcal{A}(\mathcal{B})$ by e for all $e \in \mathcal{E}$, we obtain the dyper- and hyperedge set of a (k, f, g) -flexible packing in \mathcal{F} . \square

The most important ingredient of this section is the following result that characterizes the arc sets of (k, f, g) -feasible packings as the intersection of two matroids.

Theorem 4.3.5. *Let $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ be a mixed hypergraph, $\mathcal{D}_{\mathcal{F}} = (V, \mathcal{A}' = \mathcal{A} \cup \mathcal{A}_{\mathcal{E}})$ its directed extension, $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$ integer valued functions and $k \in \mathbb{Z}_{\geq 0}$. Suppose that (2.4.3) and (2.4.4) are satisfied. Then the dyperedge sets of the (k, f, g) -feasible packings in $\mathcal{D}_{\mathcal{F}}$ are exactly the common independent sets of size $k(|V| - 1)$ of $M_{\mathcal{F}}^k$ and $M_{\mathcal{D}_{\mathcal{F}}}^{(k, f, g)}$.*

Proof. For the sake of simplicity, we use \mathbf{M}_1 and \mathbf{M}_2 for $M_{\mathcal{F}}^k$ and $M_{\mathcal{D}_{\mathcal{F}}}^{(k, f, g)}$, respectively. As (2.4.3) and (2.4.4) are satisfied, Corollary 2.4.1 yields that M_2 is well-defined.

First, let \mathcal{Z} be the dyperedge set of a (k, f, g) -feasible packing in $\mathcal{D}_{\mathcal{F}}$. Then the underlying undirected hypergraph of (V, \mathcal{Z}) is the union of k hyperedge-disjoint spanning hypertrees and \mathcal{Z} contains at most one dyperedge of the bundle \mathcal{A}_e for all $e \in \mathcal{E}$. It follows that \mathcal{Z} is an independent set of M_1 . As \mathcal{Z} is the dyperedge set of a packing of k spanning hyperarborescences, we have $|\mathcal{Z}| = k(|V| - 1)$. Since the packing is (k, f, g) -feasible, every vertex v is the root of at least $f(v)$ and at most $g(v)$ of the k spanning hyperarborescences of the packing. It follows that $k - g(v) \leq d_{\mathcal{Z}}^-(v) \leq k - f(v)$, so \mathcal{Z} is an independent set of M_2 .

Now let \mathcal{Z} be a common independent set of M_1 and M_2 of size $k(|V| - 1)$. Then, by $f \geq 0$, the underlying hypergraph of (V, \mathcal{Z}) is the union of k hyperedge-disjoint spanning hypertrees and $d_{\mathcal{Z}}^-(v) \leq k - f(v) \leq k$ for all $v \in V$. Let R be the multiset in V in which every vertex v in V is contained $k - d_{\mathcal{Z}}^-(v)$ times. Observe that this value is nonnegative for all $v \in V$. As $|\mathcal{Z}| = k(|V| - 1)$, we have $|R| = \sum_{v \in V} (k - d_{\mathcal{Z}}^-(v)) = k|V| - |\mathcal{Z}| = k|V| - k(|V| - 1) = k$. Also, by construction $d_{\mathcal{Z}}^-(v) = k - |R \cap v|$ for all $v \in V$. Theorem 4.1.5 therefore implies that \mathcal{Z} is the set of dyperedges of the union of k dyperedge-disjoint spanning hyperarborescences with root set R , so each $v \in V$ is the root of $k - d_{\mathcal{Z}}^-(v)$ of them. As \mathcal{Z} is an independent set of M_2 , we obtain that $k - g(v) \leq d_{\mathcal{Z}}^-(v) \leq k - f(v)$, so $f(v) \leq k - d_{\mathcal{Z}}^-(v) \leq g(v)$. Finally, as \mathcal{Z} is independent in M_1 , \mathcal{Z} contains at most one dyperedge of the bundle \mathcal{A}_e for all $e \in \mathcal{E}$. It follows that \mathcal{Z} is the dyperedge set of a (k, f, g) -feasible packing in $\mathcal{D}_{\mathcal{F}}$. \square

For the proof of Theorems 4.3.3 and 4.3.4, we further need two slightly technical lemmas. The first one shows the necessity in Theorem 4.3.3 and is also used in the proof of Theorem 4.3.4.

Lemma 4.3.2. *Let $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ be a mixed hypergraph, $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$ integer functions and $k \in \mathbb{Z}_{\geq 0}$. If there exists a (k, f, g) -flexible packing in \mathcal{F} , then (4.3.7), (4.3.8) and (4.3.9) are satisfied.*

Proof. By Lemma 4.3.1, $\mathcal{D}_{\mathcal{F}}$ has a (k, f, g) -feasible packing $\mathcal{B} = \{\mathcal{B}_i : i \in \{1, \dots, k\}\}$. For some $X \subseteq V$, let $\mathbf{s}(X)$ denote the number of hyperarborescences in \mathcal{B} whose roots are in X . Then for every vertex $v \in V$, $f(v) \leq s(v) \leq g(v)$ and hence (4.3.7) is satisfied. Let now \mathcal{P} be a subpartition of V . Let \mathcal{Z} be the dyperedge set of \mathcal{B} . By definition, \mathcal{Z} contains at most one dyperedge in \mathcal{A}_e for all $e \in \mathcal{E}$. It follows that $e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}) \geq \sum_{X \in \mathcal{P}} d_{\mathcal{Z}}^-(X)$. Since \mathcal{B} is a packing of spanning hyperarborescences, we have $d_{\mathcal{Z}}^-(X) \geq k - s(X)$ for all $X \in \mathcal{P}$. Thus $e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}) \geq k|\mathcal{P}| - s(\cup \mathcal{P})$. Since $s(V) = k$ and $f(v) \leq s(v) \leq g(v)$ for all $v \in V$, we have $s(\cup \mathcal{P}) = s(V) - s(V - \cup \mathcal{P}) \leq k - f(V - \cup \mathcal{P})$, yielding (4.3.8). Further, we have $s(\cup \mathcal{P}) \leq g(\cup \mathcal{P})$ yielding (4.3.9). \square

The second one allows us to use Theorem 4.3.5 in the proofs of Theorems 4.3.3 and 4.3.4.

Lemma 4.3.3. *Let $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ be a mixed hypergraph, $\mathcal{D}_{\mathcal{F}} = (V, \mathcal{A}' = \mathcal{A} \cup \mathcal{A}_{\mathcal{E}})$ its directed extension, $f, g : V \rightarrow \mathbb{Z}_{\geq 0}$ integer functions and $k \in \mathbb{Z}_{\geq 0}$. If (4.3.7), (4.3.8) and (4.3.9) are satisfied, then (2.4.3) and (2.4.4) are satisfied.*

Proof. By (4.3.8) for $\mathcal{P} = \emptyset$ and $f \geq 0$, we obtain

$$k = -k(|\emptyset| - 1) \geq f(V - \emptyset) - e_{\mathcal{E} \cup \mathcal{A}}(\emptyset) = f(V).$$

By (4.3.9) for $\mathcal{P} = \{V\}$, we obtain

$$k = k|\{V\}| \leq g(V) + e_{\mathcal{E} \cup \mathcal{A}}(\{V\}) = g(V).$$

This yields $f(V) \leq k \leq g(V)$.

Now let $v \in V$. By (4.3.7), we obtain $k - g(v) \leq k - f(v)$. By $f(V) \leq k$ and $f \geq 0$, we obtain $0 \leq k - f(V) \leq k - f(v)$. By (4.3.9) for $\mathcal{P} = \{v\}$, we obtain

$$k - g(v) = k|\mathcal{P}| - g(\cup \mathcal{P}) \leq e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}) = e_{\mathcal{E} \cup \mathcal{A}}(\{v\}) = d_{\mathcal{A}'}^-(v).$$

Then, by $0 \leq d_{\mathcal{A}'}^-(v)$, (2.4.3) follows.

Let $V' = \{v \in V : k - g(v) > 0\}$. We obtain

$$\begin{aligned} \sum_{v \in V} \max\{k - g(v), 0\} &= \sum_{v \in V'} (k - g(v)) \\ &= k|V'| - g(V') \\ &= k(|V| - 1) - (g(V') + k(|V - V'| - 1)). \end{aligned}$$

If $V' \neq V$, by $g \geq 0$ we obtain $g(V') + k(|V - V'| - 1) \geq g(V') \geq 0$. If $V' = V$, by $g(V) \geq k$, we obtain $g(V') + k(|V - V'| - 1) = g(V) - k \geq 0$. This yields the first inequality of (2.4.4).

Let $V'' = \{v \in V : k - f(v) < d_{\mathcal{A}'}^-(v)\}$. Let $\mathcal{P} = \{\{v\} : v \in V - V''\}$. Note that any element in $\mathcal{E}(\mathcal{P}) \cup \mathcal{A}(\mathcal{P})$ provides at least one dyperedge in \mathcal{A}' entering a vertex in $V - V''$. Hence, by (4.3.8), we obtain

$$\begin{aligned} \sum_{v \in V} \min\{k - f(v), d_{\mathcal{A}'}^-(v)\} &= \sum_{v \in V''} (k - f(v)) + \sum_{v \in V - V''} d_{\mathcal{A}'}^-(v) \\ &\geq (k|V''| - f(V'')) + e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}) \\ &\geq (k|V''| - f(V'')) + k(|\mathcal{P}| - 1) + f(V - \cup \mathcal{P}) \\ &= (k|V''| - f(V'')) + k(|V - V''| - 1) + f(V'') \\ &= k(|V| - 1). \end{aligned}$$

It follows that the second inequality of (2.4.4) is satisfied. \square

4.3.3 Proof of Theorem 4.3.3

We are now ready to show how Theorem 4.3.5 and Theorem 2.4.1 imply Theorem 4.3.3.

Proof. (of Theorem 4.3.3)

Necessity is proved in Lemma 4.3.2.

To see sufficiency, suppose that (4.3.7), (4.3.8) and (4.3.9) are satisfied. Let $\mathcal{D}_{\mathcal{F}} = (V, \mathcal{A}' = \mathcal{A} \cup \mathcal{A}_{\mathcal{E}})$ be the directed extension of \mathcal{F} . By Lemma 4.3.3, (2.4.3) and (2.4.4) are satisfied. It now follows from Corollary 2.4.1 that $M_{\mathcal{D}_{\mathcal{F}}}^{(k,f,g)}$ is well-defined. Again, we use M_1 and M_2 for $M_{\mathcal{F}}^k$ and $M_{\mathcal{D}_{\mathcal{F}}}^{(k,f,g)}$, respectively. Suppose for a contradiction that no (k, f, g) -flexible packing of mixed hyperarborescences exists in \mathcal{F} . By Lemma 4.3.1, no (k, f, g) -feasible packing of hyperarborescences exists in $\mathcal{D}_{\mathcal{F}}$. Now Theorem 4.3.5 implies that M_1 and M_2 do not have a common independent set of size $k(|V| - 1)$. The next result allows us to fix a certain structure leading to a contradiction later.

Claim 4.3.1. *There exist a partition \mathcal{P} of V and $\mathcal{K} \subseteq \mathcal{E}(\mathcal{P})$ such that*

$$k(|\mathcal{P}| - 1) > |\mathcal{K}| + r_{M_2}(\mathcal{A}(\mathcal{P}) \cup \mathcal{A}_{\mathcal{E}(\mathcal{P})-\mathcal{K}}). \quad (4.3.10)$$

Proof. As M_1 and M_2 do not have a common independent set of size $k(|V| - 1)$, Theorem 2.4.1 implies that there exists a dyperedge set $\mathcal{Z}' \subseteq \mathcal{A}'$ such that $k(|V| - 1) > r_{M_1}(\mathcal{Z}') + r_{M_2}(\mathcal{A}' - \mathcal{Z}')$. By Proposition 2.4.6, there exists a partition \mathcal{P} of V such that, for $\mathcal{K} = \{e \in \mathcal{E}(\mathcal{P}) : \mathcal{Z}' \cap \mathcal{A}_e \neq \emptyset\}$, we have $r_{M_1}(\mathcal{Z}') = |\mathcal{Z}' \cap \mathcal{A}(\mathcal{P})| + |\mathcal{K}| + k(|V| - |\mathcal{P}|)$. By Proposition 2.4.1 and $\mathcal{A}_{\mathcal{E}(\mathcal{P})-\mathcal{K}} \cap \mathcal{Z}' = \emptyset$, we have

$$\begin{aligned} |\mathcal{Z}' \cap \mathcal{A}(\mathcal{P})| + r_{M_2}(\mathcal{A}' - \mathcal{Z}') &\geq r_{M_2}(\mathcal{Z}' \cap \mathcal{A}(\mathcal{P})) + r_{M_2}(\mathcal{A}' - \mathcal{Z}') \\ &\geq r_{M_2}((\mathcal{Z}' \cap \mathcal{A}(\mathcal{P})) \cup (\mathcal{A}' - \mathcal{Z}')) \\ &= r_{M_2}(\mathcal{A}' - (\mathcal{Z}' - \mathcal{A}(\mathcal{P}))) \\ &\geq r_{M_2}(\mathcal{A}(\mathcal{P}) \cup \mathcal{A}_{\mathcal{E}(\mathcal{P})-\mathcal{K}}). \end{aligned}$$

The above three inequalities yield

$$\begin{aligned} k(|\mathcal{P}| - 1) &> r_{M_1}(\mathcal{Z}') + r_{M_2}(\mathcal{A}' - \mathcal{Z}') - k(|V| - |\mathcal{P}|) \\ &= |\mathcal{Z}' \cap \mathcal{A}(\mathcal{P})| + |\mathcal{K}| + r_{M_2}(\mathcal{A}' - \mathcal{Z}') \\ &\geq |\mathcal{K}| + r_{M_2}(\mathcal{A}(\mathcal{P}) \cup \mathcal{A}_{\mathcal{E}(\mathcal{P})-\mathcal{K}}). \end{aligned}$$

□

Let \mathcal{P} be the partition of V and \mathcal{K} the hyperedge set from Claim 4.3.1 and let $\mathcal{Z} = \mathcal{A}(\mathcal{P}) \cup \mathcal{A}_{\mathcal{E}(\mathcal{P})-\mathcal{K}}$. For some $X \in \mathcal{P}$, a dyperedge $a \in \mathcal{A} \cup \mathcal{A}_{\mathcal{E}-\mathcal{K}}$ contributes to either of $\sum_{v \in X} d_{\mathcal{Z}}^-(v)$ and $d_{\mathcal{A} \cup \mathcal{A}_{\mathcal{E}-\mathcal{K}}}^-(X)$ if and only if $\text{head}(a) \in X$ and $\text{tail}(a) - X \neq \emptyset$. This yields

$$\sum_{v \in X} d_{\mathcal{Z}}^-(v) = d_{\mathcal{A} \cup \mathcal{A}_{\mathcal{E}-\mathcal{K}}}^-(X). \quad (4.3.11)$$

For all $\mathcal{P}' \subseteq \mathcal{P}$, as every hyperedge in $\mathcal{E}(\mathcal{P}')$ contributes to one of $|\mathcal{K}|$ and $\sum_{X \in \mathcal{P}'} d_{\mathcal{A}_{\mathcal{E}-\mathcal{K}}}^-(X)$, we have

$$|\mathcal{K}| + \sum_{X \in \mathcal{P}'} d_{\mathcal{A} \cup \mathcal{A}_{\mathcal{E}-\mathcal{K}}}^-(X) \geq e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}'). \quad (4.3.12)$$

Using Corollary 2.4.1, we distinguish two cases depending on where the rank of \mathcal{Z} in M_2 is attained.

Case 4.3.1. $r_{M_2}(\mathcal{Z}) = \sum_{v \in V} \min\{k - f(v), d_{\mathcal{Z}}^-(v)\}$.

Let $\mathcal{P}' = \{X \in \mathcal{P} : d_{\bar{\mathcal{Z}}}^-(v) \leq k - f(v) \text{ for all } v \in X\}$. For all $X \in \mathcal{P}'$, by the definition of \mathcal{P}' and (4.3.11), we have

$$\sum_{v \in X} \min\{k - f(v), d_{\bar{\mathcal{Z}}}^-(v)\} = \sum_{v \in X} d_{\bar{\mathcal{Z}}}^-(v) = d_{\mathcal{A} \cup \mathcal{A}_{\mathcal{E}-\mathcal{K}}}^-(X).$$

For all $X \in \mathcal{P} - \mathcal{P}'$, there exists a vertex $v_X \in X$ with $k - f(v_X) < d_{\bar{\mathcal{Z}}}^-(v_X)$, and then, by $f, k - f, d_{\bar{\mathcal{Z}}}^- \geq 0$, we have

$$\begin{aligned} k - f(X) &\leq k - f(v_X) \\ &= \min\{k - f(v_X), d_{\bar{\mathcal{Z}}}^-(v_X)\} \\ &\leq \sum_{v \in X} \min\{k - f(v), d_{\bar{\mathcal{Z}}}^-(v)\}. \end{aligned}$$

Then, by (4.3.10), the case distinction made, (4.3.12) and (4.3.8) for \mathcal{P}' , we obtain

$$\begin{aligned} k(|\mathcal{P}| - 1) &> |\mathcal{K}| + r_{M_2}(\mathcal{A}(\mathcal{P}) \cup \mathcal{A}_{\mathcal{E}(\mathcal{P})-\mathcal{K}}) \\ &= |\mathcal{K}| + \sum_{v \in V} \min\{k - f(v), d_{\bar{\mathcal{Z}}}^-(v)\} \\ &= |\mathcal{K}| + \sum_{X \in \mathcal{P}'} \sum_{v \in X} \min\{k - f(v), d_{\bar{\mathcal{Z}}}^-(v)\} + \sum_{X \in \mathcal{P} - \mathcal{P}'} \sum_{v \in X} \min\{k - f(v), d_{\bar{\mathcal{Z}}}^-(v)\} \\ &\geq |\mathcal{K}| + \sum_{X \in \mathcal{P}'} d_{\mathcal{A} \cup \mathcal{A}_{\mathcal{E}-\mathcal{K}}}^-(X) + \sum_{X \in \mathcal{P} - \mathcal{P}'} (k - f(X)) \\ &\geq e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}') + k(|\mathcal{P}| - |\mathcal{P}'|) - f(V - \cup \mathcal{P}') \\ &\geq k(|\mathcal{P}| - |\mathcal{P}'|) + k(|\mathcal{P}'| - 1) \\ &= k(|\mathcal{P}| - 1), \end{aligned}$$

a contradiction.

Case 4.3.2. $r_{M_2}(\mathcal{Z}) = k(|V| - 1) - \sum_{v \in V} \max\{0, k - g(v) - d_{\bar{\mathcal{Z}}}^-(v)\}$.

Let $\mathcal{P}'' = \{X \in \mathcal{P} : k - g(v) - d_{\bar{\mathcal{Z}}}^-(v) \geq 0 \text{ for all } v \in X\}$. For all $X \in \mathcal{P}''$, by the definition of \mathcal{P}'' and (4.3.11), we have

$$\begin{aligned} \sum_{v \in X} \max\{0, k - g(v) - d_{\bar{\mathcal{Z}}}^-(v)\} &= \sum_{v \in X} k - g(v) - d_{\bar{\mathcal{Z}}}^-(v) \\ &= k|X| - g(X) - d_{\mathcal{A} \cup \mathcal{A}_{\mathcal{E}-\mathcal{K}}}^-(X). \end{aligned}$$

For all $X \in \mathcal{P} - \mathcal{P}''$, there exists a vertex $v_X \in X$ with $0 > k - g(v_X) - d_{\bar{\mathcal{Z}}}^-(v_X)$, and then, by $g, d_{\bar{\mathcal{Z}}}^- \geq 0$, we have

$$\begin{aligned} \sum_{v \in X} \max\{0, k - g(v) - d_{\bar{\mathcal{Z}}}^-(v)\} &= \sum_{v \in X - v_X} \max\{0, k - g(v) - d_{\bar{\mathcal{Z}}}^-(v)\} \\ &\leq k(|X| - 1). \end{aligned}$$

By (4.3.10) and the case distinction we made, we obtain

$$\begin{aligned} k(|\mathcal{P}| - 1) &> |\mathcal{K}| + r_{M_2}(\mathcal{Z}) \\ &= |\mathcal{K}| + k(|V| - 1) - \sum_{v \in V} \max\{0, k - g(v) - d_{\bar{\mathcal{Z}}}^-(v)\}. \end{aligned}$$

This yields

$$\begin{aligned}
k(|V| - |\mathcal{P}|) + |\mathcal{K}| &< \sum_{v \in V} \max\{0, k - g(v) - d_{\bar{z}}^-(v)\} \\
&= \sum_{X \in \mathcal{P}''} \sum_{v \in X} \max\{0, k - g(v) - d_{\bar{z}}^-(v)\} + \sum_{X \in \mathcal{P} - \mathcal{P}''} \sum_{v \in X} \max\{0, k - g(v) - d_{\bar{z}}^-(v)\} \\
&\leq \sum_{X \in \mathcal{P}''} (k|X| - g(X) - d_{\mathcal{A} \cup \mathcal{A}_{\mathcal{E} - \mathcal{K}}}^-(X)) + \sum_{X \in \mathcal{P} - \mathcal{P}''} k(|X| - 1) \\
&= k(|V| - |\mathcal{P} - \mathcal{P}''|) - \sum_{X \in \mathcal{P}''} (g(X) + d_{\mathcal{A} \cup \mathcal{A}_{\mathcal{E} - \mathcal{K}}}^-(X)).
\end{aligned}$$

We obtain by (4.3.12) and (4.3.9) for \mathcal{P}'' that

$$\begin{aligned}
k|\mathcal{P}''| &> |\mathcal{K}| + \sum_{X \in \mathcal{P}''} d_{\mathcal{A} \cup \mathcal{A}_{\mathcal{E} - \mathcal{K}}}^-(X) + \sum_{X \in \mathcal{P}''} g(X) \\
&\geq e_{\mathcal{A} \cup \mathcal{E}}(\mathcal{P}'') + g(\cup \mathcal{P}'') \\
&\geq k|\mathcal{P}''|,
\end{aligned}$$

a contradiction.

The case distinction is complete which finishes the proof of Theorem 4.3.3. \square

4.3.4 Algorithmic aspects

We here deal with the algorithmic aspects of flexible packings of mixed hyperarborescences. As matroid intersection is already used to prove Theorem 4.3.3, we obtain Theorem 4.3.4 as a simple corollary.

Proof. (of Theorem 4.3.4) It can be checked efficiently whether (2.4.3) and (2.4.4) are satisfied. If not, then, by Lemma 4.3.3, one of (4.3.7), (4.3.8) and (4.3.9) is not satisfied. By Lemma 4.3.2, no (k, f, g) -flexible packing of mixed hyperarborescences exists in \mathcal{F} . Otherwise, by Theorem 4.3.5, the common independent sets of $M_{\mathcal{F}}^k$ and $M_{\mathcal{D}_{\mathcal{F}}}^{(k, f, g)}$ of size $k(|V| - 1)$ are exactly the dyperedge sets of the (k, f, g) -feasible packings in $\mathcal{D}_{\mathcal{F}}$.

Define $\mathbf{w}' : \mathcal{A} \cup \mathcal{A}_{\mathcal{E}} \rightarrow \mathbb{R}$ by $w'(a) = w(a)$ for all $a \in \mathcal{A}$ and $w'(a) = w(e)$ for all $a \in \mathcal{A}_e$ for all $e \in \mathcal{E}$. We first check if there is a common independent set of $M_{\mathcal{F}}^k$ and $M_{\mathcal{D}_{\mathcal{F}}}^{(k, f, g)}$ of size $k(|V| - 1)$ and if this is the case, we find a common independent set \mathcal{A}^* of $M_{\mathcal{F}}^k$ and $M_{\mathcal{D}_{\mathcal{F}}}^{(k, f, g)}$ of size $k(|V| - 1)$ minimizing $w'(\mathcal{A}^*)$. This can be done in polynomial time using Theorem 2.4.2 because polynomial time independence oracles for $M_{\mathcal{F}}^k$ and $M_{\mathcal{D}_{\mathcal{F}}}^{(k, f, g)}$ are available by Lemmas 2.4.4 and 2.4.2. Now consider the subdypergraph $\mathcal{D}^* = (V, \mathcal{A}^*)$ of $\mathcal{D}_{\mathcal{F}}$ and let \mathbf{R} be the multiset in V in which every $v \in V$ is contained $k - d_{\mathcal{A}^*}^-(v)$ times. By Theorem 4.1.14, \mathcal{A}^* can be decomposed into the dyperedge set of a packing of spanning hyperarborescences with root set R in polynomial time. Replacing dyperedges in \mathcal{A}_e by e for all $e \in \mathcal{E}$, we obtain, by Lemma 4.3.1, the desired packing in \mathcal{F} . \square

4.4 Packing k -safe arborescences and related objects

This section is based on [16]. We generalize the concept of Theorem 4.1.1 in a way which is in some way the inverse of what is done in Sections 4.2 and 4.3. Instead of relaxing some of the conditions on the arborescences, we impose an additional one on them which we call being k -safe. Moreover, while most of the techniques used in Sections 4.2 and 4.3 yield characterizations, the results in this section are purely algorithmic. We first deal with packing k -safe r -arborescences and then with two objects with similar properties called (r, k) -safe flow branchings and (r, k) -safe spanning trees.

The notion of k -safe arborescences was introduced by Bang-Jensen and Yeo in [8]. For an r -arborescence $X = (V + r, A)$ and some $v \in V$, we use \mathbf{B}_X^v for the subgraph induced on the vertices reachable from v . We say that X is k -safe if $n - |V(\mathbf{B}_X^v)| \geq k$ for all $v \in V$ where $n = |V + r|$. Notice that it is enough that only the out-neighbours of r satisfy this latter condition for X to be k -safe. Bang-Jensen and Yeo prove the following negative result showing that in general not even the problem of finding a single k -safe spanning r -arborescence is tractable.

Theorem 4.4.1 (Bang-Jensen and Yeo [8]). *Given a singularly rooted digraph $D = (V \cup r, A)$, deciding whether D has an $(n - k)$ -safe spanning r -arborescence is NP-complete for any fixed $k \geq 3$.*

In this light, a characterization in the shape of Theorem 4.1.1 clearly seems out of reach. It nevertheless remains interesting to investigate the possibility of finding arc-disjoint k -safe spanning r -arborescences for small values of k . This question has been dealt with by Bang-Jensen, Havet and Yeo [6]. On the negative side, they implicitly prove the following result which shows that there is little hope to algorithmically find a packing of $k(n)$ -safe spanning r -arborescences if k is a function that does not grow too slowly. While a polynomial-time algorithm for the problem they consider would not imply $P=NP$, it would imply the failure of the Exponential Time Hypothesis.

Theorem 4.4.2. *Suppose that the ETH holds, let $\varepsilon > 0$ be arbitrary and let $k : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ be a function such that $(\log(n))^{1+\varepsilon} \leq k(n) \leq \frac{n}{2}$ for all $n > 0$. Further, suppose that there exists a constant C^* such that for all $c \geq C^*$ there exists an n such that $k(n) = c$. Then there is no algorithm running in time $n^{O(1)}$ for deciding whether a given singularly rooted digraph $D = (V \cup r, A)$ has two arc-disjoint $k(n)$ -safe spanning r -arborescences.*

On the positive side, they show that the problem becomes tractable when fixing the value of k . While several results considered in this section hold for finding an arbitrary number of disjoint objects, we focus on the case where we want to find just two of them in order to avoid technicalities. In Section 4.4.6 we discuss the generalization to more than two objects.

Theorem 4.4.3 ([6]). *Deciding whether a given singularly rooted digraph $D = (V \cup r, A)$ contains two arc-disjoint k -safe spanning r -arborescences is XP with parameter k .*

Our first contribution is to improve Theorem 4.4.3 by showing that the problem is *fixed-parameter tractable* (FPT).

Theorem 4.4.4. *Deciding whether a given singularly rooted digraph $D = (V \cup r, A)$ contains two arc-disjoint k -safe spanning r -arborescences is FPT with parameter k . More precisely, it can be solved in time $2^{O(k^2 \cdot \log k)} \cdot n^{O(1)}$. Further, if they exist, the two arc-disjoint k -safe spanning r -arborescences can be computed within the same running time.*

The second structure we consider is called (r, k) -flow branchings. It builds a connection between the theory of packing spanning arborescences and flow problems. It was introduced by Bang-Jensen and Bessy in [3].

A *flow* in a digraph $D = (V, A)$ is a function $z : A \rightarrow \mathbb{Z}_{\geq 0}$. Given a singularly rooted digraph $X = (V \cup r, A)$ and a capacity function $\mathbf{c} : A \rightarrow \mathbb{Z}_{\geq 0}$, an (r, \mathbf{c}) -*branching flow* is a flow $z : A \rightarrow \mathbb{Z}_{\geq 0}$ such that $z(a) \leq \mathbf{c}(a)$ for all $a \in A$, $z(\delta^+(r)) - z(\delta^-(r)) = n - 1$ and $z(\delta^-(v)) - z(\delta^+(v)) = 1$ for all $v \in V$. If X admits an (r, \mathbf{c}) -branching flow, we say that D is an (r, \mathbf{c}) -*flow branching*. If for some positive integer k , we have $\mathbf{c}(a) = n - k$ for all $a \in A$, we speak of an (r, k) -*branching flow* and an (r, k) -*flow branching*. Given a digraph D , an (r, k) -branching flow X that is a subgraph of D is *spanning* in D if it has the same vertex set as D .

If a singularly rooted digraph $D = (V \cup r, A)$ admits a k -safe spanning r -arborescence, then it is easy to see that D is an (r, k) -flow branching, but the converse is not necessarily true, as pointed out in [6].

Bang-Jensen and Bessy [3] consider the problem of finding arc-disjoint spanning flow branchings in a singularly rooted digraph. Among others, they show the following negative result that makes a characterization in the shape of Theorem 4.1.1 seem out of reach.

Theorem 4.4.5. *Given a singularly rooted digraph $D = (V \cup r, A)$ and a capacity function $\mathbf{c} : A \rightarrow \mathbb{Z}_{\geq 0}$, it is NP-complete to decide whether D contains two arc-disjoint spanning (r, \mathbf{c}) -flow branchings even if $\mathbf{c}(a) \in \{1, 2\}$ for all $a \in A$.*

The above result has been strengthened in [6].

Theorem 4.4.6. *Given a singularly rooted digraph $D = (V \cup r, A)$ and a fixed positive integer $k \geq 2$, it is NP-complete to decide whether D contains two arc-disjoint spanning $(r, n - k)$ -flow branchings.*

On the other hand, the problem of finding arc-disjoint spanning (r, k) -flow branchings for some small values of k turns out to be more tractable. The study of such spanning flow branchings has surprisingly many similarities with the study of k -safe spanning arborescences.

On the negative side, the following result is proven in [6]. It shows that there is little hope to algorithmically find arc-disjoint spanning $(r, k(n))$ -flow branchings if k is a function that does not grow too slowly. It can be viewed as an analogue of Theorem 4.4.2 for (r, k) -flow branchings.

Theorem 4.4.7 ([6]). *Suppose that ETH holds, let $\varepsilon > 0$ be arbitrary and let $k : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ be a function such that $(\log(n))^{1+\varepsilon} \leq k(n) \leq \frac{n}{2}$ for all $n > 0$. Further, suppose that there exists a constant C^* such that for all $c \geq C^*$ there exists an n such that $k(n) = c$. Then there is no algorithm running in time $n^{O(1)}$ for deciding whether a given singularly rooted digraph $D = (V \cup r, A)$ has two arc-disjoint spanning $(r, k(n))$ -flow branchings.*

On the positive side, Bang-Jensen and Bessy [3] showed that the case $k = 1$ can be solved in polynomial time. This result was again generalized by Bang-Jensen, Havet and Yeo [6] who proved that the problem can be solved in polynomial time for every fixed value of $k \geq 1$. The following result can be viewed as an analogue of Theorem 4.4.3 for (r, k) -flow branchings.

Theorem 4.4.8 (Bang-Jensen, Havet and Yeo [6]). *Deciding whether a given singularly rooted digraph $D = (V \cup r, A)$ contains two arc-disjoint spanning (r, k) -flow branchings is XP with parameter k .*

The authors of [6] ask whether the above problem is FPT. Our second contribution is an affirmative answer to this question.

Theorem 4.4.9. *Deciding whether a given singularly rooted digraph $D = (V \cup r, A)$ contains two arc-disjoint spanning (r, k) -flow branchings is FPT with parameter k . More precisely, it can be solved in time $2^{O(k^2 \cdot \log k)} \cdot n^{O(1)}$. Further, if they exist, the two arc-disjoint spanning (r, k) -flow branchings can be computed within the same running time.*

Finally, we consider a similar problem in undirected graphs that has also been introduced in [6]. The object we deal with is called (r, k) -safe spanning trees. Given a rooted tree $T = (V + r, E)$ and some $v \in V$, we use C_T^v for the subgraph of $T - v$ that arises from deleting the component of $T - v$ containing r . We say that T is (r, k) -safe if for every $v \in V$, we have $|V - V(C_T^v)| \geq k$. In the same way as problems on packing k -safe r -arborescences generalize Theorem 4.1.1, (r, k) -safe spanning trees generalize Theorem 2.1.5. The following result has been proven by Bang-Jensen, Havet and Yeo in [6].

Theorem 4.4.10 (Bang-Jensen, Havet, and Yeo [6]). *Deciding whether a given singularly rooted graph $G = (V \cup r, E)$ contains two edge-disjoint (r, k) -safe spanning trees is XP with parameter k .*

Again, we improve Theorem 4.4.10 as follows.

Theorem 4.4.11. *Deciding whether a given singularly rooted graph $G = (V \cup r, E)$ contains two edge-disjoint (r, k) -safe spanning trees is FPT with parameter k . More precisely, it can be solved in time $2^{O(k^2 \cdot \log k)} \cdot n^{O(1)}$. Further, if they exist, the two edge-disjoint (r, k) -safe spanning trees can be computed within the same running time.*

Since a hardness result in the spirit of Theorem 4.4.2 and Theorem 4.4.7 was not provided in [6], we fill this gap and prove the following theorem whose proof is inspired by the one of [6, Theorem 5.2]. It shows that the problem is hard even if we want to find one single (r, k) -safe spanning tree.

Theorem 4.4.12. *Let $p \geq 1$ be a fixed positive integer. Deciding whether a given singularly rooted graph $G = (V \cup r, E)$ has p edge-disjoint (r, k) -safe spanning trees is NP-complete. Moreover, let $\varepsilon > 0$ be arbitrary and let $k : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ be a function such that $(\log(n))^{2+\varepsilon} \leq k(n) \leq \frac{n}{2}$ for all $n > 0$. Further, suppose that there exists a constant C^* such that for all $c \geq C^*$ there exists an n such that $k(n) = c$. Then, assuming the ETH, there is no algorithm running in time $n^{O(1)}$ for deciding whether a given singularly rooted graph contains p edge-disjoint (r, k) -safe spanning trees.*

We next give a rough description of the technique which is used to prove Theorems 4.4.4, 4.4.9 and 4.4.11.

In order to obtain the FPT algorithms for the three considered problems, we follow a common strategy. In a nutshell, the main ideas used in the XP algorithms of [6] and how we manage to improve them to FPT algorithms can be summarized as follows. The algorithms of Bang-Jensen, Havet and Yeo [6] are all based on proving the following general property for each of the considered problems where all the substructures are (singularly) rooted:

A given (di)graph contains the required substructure \mathcal{X} (i.e., a pair of disjoint spanning arborescences, flow branchings, or spanning trees) if and only if it contains another type of substructure \mathcal{X}' of size bounded by a function of k and such that, if found, it can be extended to the required substructure \mathcal{X} in polynomial time.

Once the above property is proved, an XP algorithm follows naturally: generate all candidate substructures \mathcal{X}' in time $n^{f(k)}$ and, for each of them, try to extend it to a substructure \mathcal{X} in polynomial time. Our main contribution is to prove that the above general property is still true if we replace \mathcal{X}' with another type of substructure \mathcal{X}'' having the crucial

property that the candidate substructures \mathcal{X}'' can be all enumerated in time $f(k) \cdot n^{O(1)}$, hence yielding an FPT algorithm. In order to achieve this, we prove that we can restrict ourselves to objects \mathcal{X}'' whose “non-sink” vertices (i.e., those with positive (out-)degree in \mathcal{X}'') have *(out-)degree, in the original (di)graph, bounded by some function of k , namely $O(k)$ or $O(k^2)$* . Intuitively, this is possible because, given a pair $\mathcal{X}' = \{X_1, X_2\}$ containing a vertex v of large (out-)degree (as a function of k) in, say, X_1 , we can safely prune the “branch” of X_1 hanging from v , with the guarantee that it will always be possible to extend the pruned substructure to another substructure of the original type. Note that since the substructures \mathcal{X}'' have size *and* maximum (out-)degree bounded by a function of k , we can indeed generate all candidate substructures in time $f(k) \cdot n^{O(1)}$, as required.

Let us first focus on the problem of finding two arc-disjoint k -safe spanning r -arborescences. In this case, the substructure \mathcal{X}' is an *extendable pair of arc-disjoint classic (r, k) -kernels*. In Theorem 4.4.3, we restate a result from [6] that shows that the existence of this substructure is sufficient for the existence of two arc-disjoint k -safe spanning r -arborescences. We then introduce compact (r, k) -kernels. An *extendable pair of arc-disjoint compact (r, k) -kernels* corresponds to the substructure \mathcal{X}'' . In Lemma 4.4.10, we show that the existence of an extendable pair of arc-disjoint compact (r, k) -kernels is also sufficient for the existence of two arc-disjoint k -safe spanning r -arborescences. The proof of Lemma 4.4.10 is our main technical contribution. Having Lemma 4.4.10 at hand, the proof of Theorem 4.4.4 is easy.

As for packing (r, k) -flow branchings, the substructure \mathcal{X}' defined in [6] is an *extendable pair of arc-disjoint classic (r, k) -cores*. In Theorem 4.4.7, we restate a result from [6] that shows that the existence of this substructure is sufficient for the existence of two arc-disjoint (r, k) -flow branchings. We then introduce compact (r, k) -cores. An *extendable pair of arc-disjoint compact (r, k) -cores* corresponds to the substructure \mathcal{X}'' . In Lemma 4.4.11, we show that the existence of an extendable pair of arc-disjoint compact (r, k) -cores is also sufficient for the existence of two arc-disjoint (r, k) -flow branchings. Again, the proof of Lemma 4.4.11 is our main technical contribution and is similar to the one of Lemma 4.4.10. Again, having Lemma 4.4.11 at hand, the proof of Theorem 4.4.9 is easy.

Finally, for packing (r, k) -safe spanning trees, the substructure \mathcal{X}' defined in [6] is a *completable pair of edge-disjoint classic (r, k) -certificates*. In Theorem 4.4.9, we restate a result from [6] that shows that the existence of this substructure is sufficient for the existence of two arc-disjoint (r, k) -safe spanning trees. We then introduce compact (r, k) -certificates. A *completable pair of edge-disjoint compact (r, k) -certificates* corresponds to the substructure \mathcal{X}'' . In Lemma 4.4.12, we show that the existence of a completable pair of edge-disjoint compact (r, k) -certificates is also sufficient for the existence of two edge-disjoint (r, k) -safe spanning trees. Again, the proof of Lemma 4.4.12 is our main technical contribution and is similar to the one of Lemma 4.4.10 and Lemma 4.4.11. Again, having Lemma 4.4.12 at hand, the proof of Theorem 4.4.11 is easy.

In Section 4.4.1, we give some preliminary results for our work. In Sections 4.4.2, 4.4.3, 4.4.4, and 4.4.5 we give the proof of Theorems 4.4.4, 4.4.9, 4.4.11, and 4.4.12, respectively. Finally, we conclude our work on balanced structures in Section 4.4.6.

4.4.1 Preliminaries

In this section we collect some more technical previous results and prove several preliminary statements. We first give some general results on graphs and digraphs and then some which are more specific to each of the particular applications.

4.4.1.1 General preliminaries

Given a 2-root-connected rooted digraph $D = (V \cup r, A)$, a pair of subgraphs $(X_1 = (V_1 \cup r, A_1), X_2 = (V_2 \cup r, A_2))$ of D is called *extendable* if both $D - A_1$ and $D - A_2$ are root-connected. The following is an immediate consequence of the fact that checking whether a digraph is root-connected can clearly be done in polynomial time.

Lemma 4.4.1. *Given a rooted digraph $D = (V \cup r, A)$ and a pair of two subgraphs $(X_1 = (V_1 \cup r, A_1), X_2 = (V_2 \cup r, A_2))$, we can decide in polynomial time whether (X_1, X_2) is extendable.*

We now give one more result in undirected graphs.

Lemma 4.4.2. *Let $G = (V, E)$ be a graph and let T be a spanning tree of G . Let $e = uv \in E - E(T)$. Then there is some $f \in E(T)$ that is incident to u such that $(T - e) \cup f$ is a spanning tree of G .*

Proof. The graph $T \cup f$ contains a unique cycle C such that $uv \in E(C)$ and the deletion of an arbitrary edge of $E(C)$ yields a spanning tree of G . As C is a cycle, $E(C)$ contains an edge f different from e that is incident to u . This edge satisfies the condition. \square

4.4.1.2 Preliminaries on k -safe spanning r -arborescences

Given a rooted digraph $D = (V \cup r, A)$, a *classic (r, k) -kernel* is a subarborescence $X = (V' \cup r, A')$ of D such that X is k -safe and $|V'| = 2k - 2$. The XP algorithm of 4.4.3 is based on the following result, which we reformulate here using our terminology.

Lemma 4.4.3 (Bang-Jensen, Havet, and Yeo [6]). *Let $D = (V \cup r, A)$ be a singularly rooted digraph with $|V| \geq 2k - 2$. Then D contains two arc-disjoint k -safe spanning r -arborescences if and only if D contains an extendable pair of arc-disjoint classic (r, k) -kernels. Further, the two arc-disjoint k -safe spanning r -arborescences can be constructed from the extendable pair of classic (r, k) -kernels in polynomial time.*

4.4.1.3 Preliminaries on spanning (r, k) -flow branchings

We first need the following result that allows to recognize (r, k) -flow branchings.

Lemma 4.4.4 (Bang-Jensen, Havet, and Yeo [6]). *Given a rooted digraph $D = (V \cup r, A)$ and a non-negative integer k , we can decide in polynomial time whether D is an (r, k) -flow branching.*

Given a digraph $D = (V, A)$ and two vertices $u, v \in V$, a *uv -path flow* is a flow \mathbf{z} such that $\mathbf{z}(a) = 1$ for all arcs $a \in A(P)$ and $\mathbf{z}(a) = 0$ for all $a \in A - A(P)$ for some uv -path P . Similarly, a *cycle flow* is a flow \mathbf{z} such that $\mathbf{z}(a) = 1$ of all $a \in A(C)$ and $\mathbf{z}(a) = 0$ for all $a \in A - A(C)$ for some cycle C . We need the following result on flows which is proven in a more general form in [5].

Lemma 4.4.5. *Let $X = (V \cup r, A)$ be a flow branching and $z : A \rightarrow \mathbb{Z}_{\geq 0}$ be a branching flow in X . Then there is an rv -path flow z_v for all $v \in V$ and a set of cycle flows $\{z_C : C \in \mathcal{C}\}$ for some set of cycles \mathcal{C} such that $z = \sum_{v \in V} z_v + \sum_{C \in \mathcal{C}} z_C$.*

We now use Lemma 4.4.5 to prove an important property of arc-minimal (r, k) -flow branchings. A singularly rooted digraph $X = (V \cup r, A)$ is called *triple-free* if it does not contain more than two arcs in the same direction between the same two vertices.

Lemma 4.4.6. *Let $X = (V \cup r, A)$ be an arc-minimal (r, k) -flow branching with $|V| \geq 2k - 1$. Then X is triple-free.*

Proof. Suppose that X contains three arcs $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ whose tail is u and whose head is v for some $u, v \in V \cup r$. Further, let $z : A \rightarrow \mathbb{Z}$ be an (r, k) -branching flow. By Lemma 4.4.5, we obtain that $z = \sum_{v \in V} z_v + \sum_{C \in \mathcal{C}} z_C$ where z_v is an rv -path flow for all $v \in V$ and z_C is a cycle flow for all $C \in \mathcal{C}$ for some set of cycles \mathcal{C} . Let $z' = \sum_{v \in V} z_v$. Observe that z' is an (r, k) -branching flow. As all of the z_v are path flows, we obtain $z'(a_1) + z'(a_2) + z'(a_3) \leq |V| < 2(n - k)$. It follows that we can define a flow $z'' : A \rightarrow \mathbb{Z}$ such that $z''(a_1) + z''(a_2) = z'(a_1) + z'(a_2) + z'(a_3)$, $z''(a_3) = 0$ and $z''(a) = z'(a)$ for all $a \in A - \{a_1, a_2, a_3\}$. It is easy to see that z'' is an (r, k) -branching flow, so $X - a_3$ is an (r, k) -flow branching, a contradiction to the minimality of X . \square

Given a singularly rooted digraph $D = (V \cup r, A)$ with $|V \cup r| \geq 2k$, a *classic* (r, k) -core is an (r, k) -flow branching $X = (V' \cup r, A')$ that is a subgraph of D with $|V'| = 2k - 1$. The XP algorithm of Theorem 4.4.8 is based on the following result, again reformulated using our terminology.

Lemma 4.4.7 (Bang-Jensen, Havet, and Yeo [6]). *Let $D = (V \cup r, A)$ be a rooted digraph with $|V| \geq 2k - 1$. Then D contains two arc-disjoint spanning (r, k) -flow branchings if and only if D contains an extendable pair of arc-disjoint classic (r, k) -cores. Further, the two arc-disjoint spanning (r, k) -flow branchings can be constructed in polynomial time from the extendable pair of arc-disjoint classic (r, k) -cores.*

4.4.1.4 Preliminaries on (r, k) -safe spanning trees

We first define an equivalent for extendability in undirected graphs. Given a singularly rooted graph $G = (V \cup r, E)$, a pair of subtrees (X_1, X_2) is called *completable*, if there are edge-disjoint spanning trees T_1, T_2 of G such that $E(X_i) \subseteq E(T_i)$. Note that we require a completable pair of trees to be edge-disjoint. This is in contrast to the fact that, in Subsections 4.4.1.2 and 4.4.1.3 we do *not* require the elements in an extendable pair to be arc-disjoint. We adopt this asymmetric choice for technical reasons arising from the proofs. The following result that allows to test completability can be established using matroid theory as mentioned in [6].

Lemma 4.4.8. *Given a graph $G = (V \cup r, E)$ and a pair of subtrees (X_1, X_2) , we can decide in polynomial time whether (X_1, X_2) is completable.*

Given a singularly rooted graph $G = (V \cup r, E)$ with $|V \cup r| \geq 2k - 1$, a *classic* (r, k) -certificate is an (r, k) -safe subtree $X = (V' \cup r, E')$ of G with $|V'| = 2k - 2$. The XP algorithm of Theorem 4.4.10 is based on the following result, again restated using our terminology.

Lemma 4.4.9 (Bang-Jensen, Havet, and Yeo [6]). *Let $G = (V \cup r, E)$ be a singularly rooted graph. Then G contains two edge-disjoint (r, k) -safe spanning trees if and only if G contains a completable pair of classic (r, k) -certificates. Further, given a completable pair of classic (r, k) -certificates, we can compute two edge-disjoint (r, k) -safe spanning trees in polynomial time.*

4.4.2 An FPT algorithm for packing k -safe spanning arborescences

This section is concerned with proving Theorem 4.4.4. Given a singularly rooted digraph $D = (V \cup r, A)$ and a positive integer k , we say that a vertex $v \in V \cup r$ is *large* if $|N_D^+(v)| \geq 6k - 5$ and *small* otherwise. We let \mathbf{L}_D (resp. \mathbf{S}_D) be the set of vertices in $V \cup r$ which are large (resp. small) in D .

We are now ready to introduce a new notion of (r, k) -kernels for k -safe spanning r -arborescences. A *compact (r, k) -kernel* is a subgraph $X = (V' \cup r, A')$ of D with $|V'| \leq 2k - 2$ satisfying the following:

- X is an r -arborescence,
- all vertices in $(V' \cup r) \cap L_D$ are sinks of X , and
- a k -safe r -arborescence Y can be obtained from X by adding a set V^* of $2k - 2 - |V'|$ new vertices and adding an arc from a vertex in $(V' \cup r) \cap L_D$ to v for all $v \in V^*$.

Observe that Y is not necessarily a subgraph of D . The following result shows that compact (r, k) -kernels can be used in a similar way as classic (r, k) -kernels.

Lemma 4.4.10. *Let $D = (V \cup r, A)$ be a 2-root-connected singularly rooted digraph with $|V| \geq 2k - 2$. Then D contains two arc-disjoint k -safe spanning r -arborescences if and only if D contains an extendable pair of arc-disjoint compact (r, k) -kernels. Further, given an extendable pair of arc-disjoint compact (r, k) -kernels, we can find a pair of arc-disjoint k -safe spanning r -arborescences in polynomial time.*

Proof. By Lemma 4.4.3, for the first part it suffices to prove that D contains a pair of arc-disjoint extendable compact (r, k) -kernels if and only if D contains a pair of arc-disjoint extendable classic (r, k) -kernels.

First let $(\mathbf{X}_1 = (\mathbf{V}_1 \cup r, \mathbf{A}_1), \mathbf{X}_2 = (\mathbf{V}_2 \cup r, \mathbf{A}_2))$ be an extendable pair of arc-disjoint classic (r, k) -kernels. Let $\mathbf{X}'_i = (\mathbf{V}'_i \cup r, \mathbf{A}'_i)$ be obtained from X_i by deleting $B_{X_i}^v - v$ for all $v \in (V_i \cup r) \cap L_D$. By construction, the X'_i are r -arborescences and all vertices in $(V'_i \cup r) \cap L_D$ are sinks in X'_i . Let \mathbf{Y}_i be obtained from X'_i by adding the vertices in $V_i - V'_i$ and adding an arc from a vertex $v \in (V'_i \cup r) \cap L_D$ to a vertex $w \in V_i - V'_i$ whenever $w \in V(B_{X_i}^v) - v$. Observe that Y_i is an arborescence with $|V(Y_i)| = 2k - 1$. Further, note that $|(V_i \cup r) - V(B_{Y_i}^v)| = |(V_i \cup r) - V(B_{X_i}^v)| \geq k$ for all $v \in V'_i$ and $|V(B_{Y_i}^v)| = 0$ for all $v \in V_i - V'_i$. This yields that $|(V_i \cup r) - V(B_{Y_i}^v)| \geq |(V_i \cup r) - V(B_{X_i}^v)| \geq k$ for all $v \in V_i$ and so Y_i is a k -safe arborescence. By definition, we obtain that (X'_1, X'_2) is a pair of arc-disjoint compact (r, k) -kernels. Further, $D - A_i$ is a subgraph of $D - A'_i$ that is root-connected, so $D - A'_i$ is root-connected as well. This yields that (X'_1, X'_2) is extendable.

Now let $(\mathbf{X}_1 = (\mathbf{V}_1 \cup r, \mathbf{A}_1), \mathbf{X}_2 = (\mathbf{V}_2 \cup r, \mathbf{A}_2))$ be an extendable pair of arc-disjoint compact (r, k) -kernels. By definition, there are k -safe arborescences $\mathbf{Y}_1, \mathbf{Y}_2$ such that Y_i is obtained from X_i by adding a set \mathbf{V}_i^* of $2k - 2 - |V_i|$ new vertices and an arc from a vertex in $V_i \cap L_D$ to v for all $v \in \mathbf{V}_i^*$. Let $(\mathbf{X}'_1 = (\mathbf{V}'_1 \cup r, \mathbf{A}'_1), \mathbf{X}'_2 = (\mathbf{V}'_2 \cup r, \mathbf{A}'_2))$ be a pair of subgraphs of D that are vertex-maximal with the following properties:

- (i) X'_i is obtained from X_i by repeatedly adding another vertex $v \in V - V_i$ and an arc of A that goes from a vertex in $V_i \cap L_D$ to v ,
- (ii) $d_{X'_i}^+(v) \leq d_{Y_i}^+(v)$ for all $v \in V_i \cap L_D$,
- (iii) A'_1 and A'_2 are disjoint, and
- (iv) (X'_1, X'_2) is extendable.

Note that (X_1, X_2) satisfies conditions (i)-(iv), so (X'_1, X'_2) is well-defined. Also observe that if condition (ii) is satisfied with equality for all $v \in V_i \cap L_D$, then X'_i is isomorphic to Y_i , so X'_i is a k -safe arborescence and by definition also a classic (r, k) -kernel. If this is the case for both X'_1 and X'_2 , we are done by conditions (iii) and (iv).

We may therefore suppose by symmetry that there is a vertex $v \in V_1 \cap L_D$ with $d_{X'_1}^+(v) < d_{Y_1}^+(v)$. For any $\mathbf{a} = vz_{\mathbf{a}}$ with $z_{\mathbf{a}} \in N_D^+(v) - V'_1$, let $\mathbf{X}_{\mathbf{a}} = (V'_1 \cup z_{\mathbf{a}}, A'_1 \cup a)$. By the maximality of X'_1 , we obtain that $X_{\mathbf{a}}$ violates one of conditions **(i)**-**(iv)** for all $a \in \delta_D^+(v)$ with $z_{\mathbf{a}} \in N_D^+(v) - V'_1$. By construction and the choice of v , $X_{\mathbf{a}}$ satisfies conditions **(i)** and **(ii)** for all $a \in \delta_D^+(v)$ with $z_{\mathbf{a}} \in N_D^+(v) - V'_1$. If $X_{\mathbf{a}}$ violates **(iii)**, then $z_{\mathbf{a}} \in V'_2$. By **(ii)**, we have $|V'_2| \leq |V_2 \cup V_2^*| = 2k - 2$ and so this is the case for at most $2k - 2$ vertices $z_{\mathbf{a}} \in N_D^+(v) - V'_1$. If $X_{\mathbf{a}}$ does not satisfy **(iv)**, then a is critical in $D - A'_1$. As D is 2-root-connected and X'_1 is an arborescence by construction, Lemma 2.2.1 implies that this is the case for at most $|A'_1| = |V'_1| \leq |V_1 \cup V_1^*| = 2k - 2$ vertices $z_{\mathbf{a}} \in N_D^+(v) - V'_1$. As $v \in L_D$ and $|V'_1| \leq |V_1 \cup V_1^*| = 2k - 2$, we have $|N_D^+(v) - V'_1| \geq (6k - 5) - (2k - 2) > (2k - 2) + (2k - 2)$, so there is at least one vertex in $z \in N_D^+(v) - V'_1$ and an arc $a = vz$ such that $X_{\mathbf{a}}$ does not violate any of conditions **(i)**-**(iv)**, a contradiction.

Observe that the second part of the proof yields an algorithm for computing a pair of arc-disjoint extendable classic (r, k) -kernels. Indeed, every time we try to add an arc $a = vz$ to X'_i , we test if $(X_{\mathbf{a}}, X'_{3-i})$ satisfies conditions **(i)**-**(iv)**. Conditions **(i)**-**(iii)** can clearly be checked in polynomial time and, by Lemma 4.4.1, condition **(iv)** can also be checked in polynomial time. Never testing an arc that is parallel to one that we have tested already, after at most $4k - 4$ failed attempts, we manage to add a new vertex to V'_i . We repeat this procedure $|V_i^*| \leq 2k - 2$ times. It follows that a pair of arc-disjoint extendable classic (r, k) -kernels can be computed in time $k^2 \cdot n^{O(1)} = n^{O(1)}$. By the second part of Lemma 4.4.3, we can then find the arc-disjoint k -safe spanning arborescences in D in time $n^{O(1)}$. Therefore, the overall running time of the algorithm is polynomial, as claimed. \square

We are now ready to proceed to the proof of Theorem 4.4.4.

Proof. (of Theorem 4.4.4)

We may suppose that there are at most two parallel arcs from u to v for any $u, v \in V$. If $|V| < 2k - 2$, the problem can be solved by a brute force algorithm in time $2^{O(k^2)}$, by generating all pairs of subgraphs of D and checking whether any of these pairs satisfies the required conditions. We may hence also suppose that $|V| \geq 2k - 2$.

We can first decide in time $n^{O(1)}$ if D is 2-root-connected. If it is not, the answer is negative, so we may suppose it is. Let $\mathbf{X}_1 = (V_1 \cup r, A_1)$, $\mathbf{X}_2 = (V_2 \cup r, A_2)$ be two subgraphs of D with $|V_1|, |V_2| \leq 2k - 2$. In order to test whether X_i is a compact (r, k) -kernel, we first verify if X_i is an r -arborescence such that all the vertices in $V_i \cap L_D$ are sinks in X_i . If this is the case, we add a set V_i^* of $2k - 2 - |V_i|$ new vertices to X_i . We then test all possibilities to add one arc from $(V_i \cup r) \cap L_D$ to v for all $v \in V_i^*$. As $|V_i \cup r| \leq 2k$ and $|V_i^*| \leq 2k$, there are at most $2k^{2k} = 2^{O(k \cdot \log k)}$ possibilities to check. For each of these possibilities, we can then check in polynomial time whether the obtained graph is a k -safe arborescence.

For each such pair X_1, X_2 , by the definition of classic (r, k) -kernels, we can therefore check in time $2^{O(k \cdot \log k)}$ whether both X_1 and X_2 are compact (r, k) -kernels. By Lemma 4.4.1, we can therefore decide in time $2^{O(k \cdot \log k)} + n^{O(1)}$ if (X_1, X_2) is an extendable pair of arc-disjoint compact (r, k) -kernels in D . By Lemma 4.4.10, it therefore suffices to prove that there are at most $2^{O(k^2 \cdot \log k)}$ possible candidates for the extendable pair of arc-disjoint compact (r, k) -kernels, and that these can be generated within this running time.

Let $X = (V' \cup r, A')$ be a compact (r, k) -kernel in D . Observe that every vertex in V' can be reached from r by a directed path all of whose interior vertices are in S_D and whose length is at most $k - 1$. As every vertex in S_D has at most $6k - 6$ out-neighbors, we obtain that the number of vertices that can be reached by such a path is at most $1 + (6k - 6) + (6k - 6)^2 + \dots + (6k - 6)^{k-1} \leq (6k)^k$. As V' contains at most $2k - 2$ vertices, there are at most $\binom{(6k)^k}{2k-2} \leq (6k)^{2k^2}$ possibilities to choose V' .

Now suppose that we have chosen V' of size $2k - 2$. As there are at most two arcs in the same direction between any two vertices, there are at most $4 \binom{|V' \cup r|}{2} \leq 16k^2$ arcs that have their head and tail in $V' \cup r$. As $|A'| = 2k - 2$, there are at most $\binom{16k^2}{2k-2} \leq (16k^2)^{2k}$ possibilities to choose A' . It follows that there are at most $(6k)^{2k^2} \cdot (16k^2)^{2k}$ possibilities to choose a compact (r, k) -kernel X . As these can be computed by a brute force method, the algorithm can finish after checking less than $f(k) = \binom{(6k)^{2k^2} \cdot (16k^2)^{2k}}{2} = 2^{O(k^2 \cdot \log k)}$ candidates for the extendable pair of arc-disjoint compact (r, k) -kernels.

If no extendable pair of arc-disjoint compact (r, k) -kernels exists, by Lemma 4.4.10, D does not contain two arc-disjoint k -safe spanning arborescences. On the other hand, once we have found an extendable pair of arc-disjoint compact (r, k) -kernels, we can compute the two arc-disjoint k -safe spanning r -arborescences in polynomial time by the second part of Lemma 4.4.10. The overall running time of the obtained algorithm is $2^{O(k^2 \cdot \log k)} \cdot n^{O(1)}$. \square

4.4.3 An FPT algorithm for packing spanning (r, k) -flow branchings

This section is concerned with proving Theorem 4.4.9. Slightly modifying the terminology introduced in Section 4.4.2, given a 2-root-connected singularly rooted digraph $D = (V \cup r, A)$ and a positive integer k , we say that a vertex v is *large* if $|N^+(v)| \geq 20k^2 + 1$, and *small* otherwise. Again, we let L_D (resp. S_D) be the set of vertices in $V \cup r$ which are large (resp. small) in D .

We are now ready to introduce a new notion of (r, k) -cores for spanning (r, k) -flow branchings. A *compact (r, k) -core* is a subgraph $X = (V' \cup r, A')$ of D with $|V'| \leq 2k - 1$ satisfying the following:

- all vertices in $(V' \cup r) \cap L_D$ are sinks in X and
- an (r, k) -flow branching Y can be obtained from X by adding a set V^* of $2k - 1 - |V'|$ new vertices and adding an arc from a vertex in $(V' \cup r) \cap L_D$ to v for all $v \in V^*$.

Observe that Y is not necessarily a subgraph of D . The following result, which is similar to Lemma 4.4.10, shows that compact (r, k) -cores can be used in a similar way as classic (r, k) -cores.

Lemma 4.4.11. *Let $D = (V \cup r, A)$ be a 2-root-connected singularly rooted digraph with $|V| \geq 2k - 1$. Then D has two arc-disjoint spanning (r, k) -flow branchings if and only if D contains an extendable pair of arc-disjoint compact (r, k) -cores. Further, given an extendable pair of triple-free arc-disjoint compact (r, k) -cores, we can find a pair of arc-disjoint spanning (r, k) -flow branchings in polynomial time.*

Proof. By Lemma 4.4.7, for the first part it suffices to prove that D contains an extendable pair of arc-disjoint compact (r, k) -cores if and only if D contains an extendable pair of arc-disjoint classic (r, k) -cores.

First let $(\mathbf{X}_1 = (\mathbf{V}_1 \cup r, \mathbf{A}_1), \mathbf{X}_2 = (\mathbf{V}_2 \cup r, \mathbf{A}_2))$ be an extendable pair of arc-disjoint classic (r, k) -cores. Let $\mathbf{X}'_i = (\mathbf{V}'_i \cup r, \mathbf{A}'_i)$ be obtained from X_i by first deleting all arcs in A_i whose tail is a large vertex and then restricting to the subgraph that is root-connected from r . As $A'_i \subseteq A_i$, (X'_1, X'_2) is extendable. It remains to show that X'_1 and X'_2 are compact (r, k) -cores. By construction, all vertices in $(V'_i \cup r) \cap L_D$ are sinks in X'_i . As X_i is an (r, k) -flow branching, there is an (r, k) -branching flow $z : A_i \rightarrow \mathbb{Z}$ in X_i . Create \mathbf{Y}_i from X_i by attaching $z(\delta_{X_i}^+(v))$ arcs directed away from v to every vertex $v \in (V'_i \cup r) \cap L_D$. Assigning $z'(a) = 1$ for all arcs leaving a large vertex in Y_i and $z'(a) = z(a)$ for all

remaining arcs, we obtain that z' is an (r, k) -branching flow in Y_i , so Y_i is an (r, k) -flow branching. Furthermore, we have $|V(Y_i) - r| = z'(\delta_{Y_i}^+(r)) = z(\delta_{X_i}^+(r)) = |V_i| = 2k - 1$. It follows by definition that X_i' is a compact (r, k) -core.

Now let $(\mathbf{X}_1 = (\mathbf{V}_1 \cup r, \mathbf{A}_1), \mathbf{X}_2 = (\mathbf{V}_2 \cup r, \mathbf{A}_2))$ be an extendable pair of arc-disjoint compact (r, k) -cores. Possibly deleting arcs, we may suppose by Lemma 4.4.6 and as $|V| \geq 2k - 1$ that X_1 and X_2 are triple-free. By definition, there are (r, k) -flow branchings $\mathbf{Y}_1, \mathbf{Y}_2$ such that Y_i is obtained from X_i by adding a set \mathbf{V}_i^* of $2k - 1 - |V_i|$ new vertices and adding an arc from a vertex in $V_i \cap L_D$ to v for all $v \in V_i^*$. Let $\mathbf{X}'_1 = (\mathbf{V}'_1 \cup r, \mathbf{A}'_1)$ and $\mathbf{X}'_2 = (\mathbf{V}'_2 \cup r, \mathbf{A}'_2)$ be subgraphs of D that are vertex-maximal with the following properties:

- (i) X_i' is obtained from X_i by repeatedly adding another vertex $v \in V - V_i$ and an arc of A that goes from a vertex in $(V_i \cup r) \cap L_D$ to v ,
- (ii) $d_{X_i'}^+(v) \leq d_{Y_i}^+(v)$ for all $v \in (V_i \cup r) \cap L_D$,
- (iii) A'_1 and A'_2 are disjoint, and
- (iv) (X'_1, X'_2) is extendable.

Note that (X_1, X_2) satisfies conditions **(i)-(iv)**, so (X'_1, X'_2) is well-defined. Further, observe that if condition **(ii)** is satisfied with equality for all $v \in (V_i \cup r) \cap L_D$, then X_i' is isomorphic to Y_i , so X_i' is an (r, k) -flow branching, thus by definition also a classic (r, k) -core. If this is the case for both X'_1 and X'_2 , we are done by conditions **(iii)** and **(iv)**.

We may therefore suppose by symmetry that there is some $v \in (V_1 \cup r) \cap L_D$ with $d_{X'_1}^+(v) < d_{Y_1}^+(v)$. For any $\mathbf{a} = v\mathbf{z}_a$ with $z_a \in N_D^+(v) - V'_1$, let $X_a = (V'_1 \cup z_a, A'_1 \cup a)$. By the maximality of X'_1 , X_a violates one of conditions **(i)-(iv)** for all $a \in \delta_D^+(v)$ with $z_a \in N_D^+(v) - V'_1$. By construction and the choice of v , X_a satisfies conditions **(i)** and **(ii)** for all $a \in \delta_D^+(v)$ with $z_a \in N_D^+(v) - V'_1$. If X_a violates condition **(iii)**, then $z_a \in V'_2$. As $|V'_2| \leq |V(Y_2)| = 2k$, this is the case for at most $2k$ vertices in $N_D^+(v) - V'_1$. If X_a does not satisfy **(iv)**, then a is critical in $D - A'_1$. As X_1 is triple-free and $|V'_i| \leq 2k$ and by construction, we obtain that $|A'_1| \leq 4 \binom{|V'_1|}{2} \leq 16k^2$. Now Lemma 2.2.1 implies that this is the case for at most $16k^2$ vertices in $N_D^+(v) - V'_1$. As $v \in L_D$ and $|V'_1| \leq |V_1 \cup V_1^*| = 2k - 1$, we have $|N_D^+(v) - V'_1| \geq (20k^2 + 1) - (2k - 1) > 2k + 16k^2$, so there is at least one vertex $z \in N_D^+(v) - V'_1$ and an arc $a = vz$ such that X_a does not violate any of conditions **(i)-(iv)**, a contradiction.

Observe that the second part of the proof yields an algorithm for computing a pair of arc-disjoint extendable classic (r, k) -cores. Indeed, every time we try to add an arc $a = vz$ to X'_i , we test if (X_a, X'_{3-i}) satisfies conditions **(i)-(iv)**. Conditions **(i)-(iii)** can clearly be checked in polynomial time and, by Lemma 4.4.1, condition **(iv)** can also be checked in polynomial time. Never checking an arc a which is parallel to an arc we have already checked, after at most $20k^2$ failed attempts, we manage to add a new vertex to V'_i . We repeat this procedure at most $|V_i^*| \leq 2k$ times. It follows that a pair of arc-disjoint extendable classic (r, k) -cores can be computed in time $40k^3 \cdot n^{O(1)} = n^{O(1)}$. By the second part of Lemma 4.4.7, we can then find the arc-disjoint spanning (r, k) -flow branchings in D in polynomial time. The overall running time of the algorithm is polynomial, as claimed. \square

We are now ready to proceed to the proof of Theorem 4.4.9.

Proof. (of Theorem 4.4.9) First consider the case that $|V| < 2k - 1$. Observe that any arc-minimal spanning (r, k) -flow branching has at most $|V|$ parallel arcs between any two

vertices. It follows that, for any two vertices u, v , at most $\gamma = 4k^2$ different distributions of the arcs between u and v among the two candidates for the spanning (r, k) -flow branchings have to be considered, including taking none of these arcs. Since there are $\mu = \binom{|V|}{2} = O(k^2)$ pairs of vertices, the total number of choices for these distributions is $\gamma^\mu = 2^{O(k^2 \cdot \log k)}$. The problem can therefore be solved by a brute force algorithm in time $2^{O(k^2 \cdot \log k)}$, by generating all $2^{O(k^2 \cdot \log k)}$ pairs of candidate subgraphs of D and checking whether any of these pairs satisfies the required conditions. We may hence suppose that $|V| \geq 2k - 1$.

By Lemma 4.4.6, we may also suppose that there are at most four parallel arcs between any two vertices in D . We can first decide in polynomial time if D is 2-root-connected. If it is not, the answer is negative, so we may suppose it is. Let $\mathbf{X}_1 = (\mathbf{V}_1 \cup r, \mathbf{A}_1)$, $\mathbf{X}_2 = (\mathbf{V}_2 \cup r, \mathbf{A}_2)$ be two subgraphs of D with $|V_1|, |V_2| \leq 2k - 1$. In order to test whether X_i is a compact (r, k) -core, we first test if all vertices in $(V_i \cup r) \cap L_D$ are sinks in X_i . We then add a set \mathbf{V}_i^* of $2k - 1 - |V_i|$ new vertices to X_i . We then test all possibilities to add one arc from $(V_i \cup r) \cap L_D$ to v for all $v \in V_i^*$. As $|V_i|, |V_i^*| \leq 2k$, there are at most $2k^{2k} = 2^{O(k \cdot \log k)}$ possibilities to check. By Lemma 4.4.4, we can check in time polynomial in k whether each of the resulting graphs is an (r, k) -flow branching. Thus, we can check in time $2^{O(k \cdot \log k)}$ whether both X_1 and X_2 are compact (r, k) -cores. By Lemma 4.4.1, we can therefore decide in time $2^{O(k \cdot \log k)} + n^{O(1)}$ if (X_1, X_2) is an extendable pair of arc-disjoint compact (r, k) -cores in D . By Lemma 4.4.11, it therefore suffices to prove that there are at most $2^{O(k^2 \cdot \log k)}$ possible candidates for the extendable pair of arc-disjoint compact (r, k) -cores, and that these can be generated within the same running time.

Let $\mathbf{X} = (\mathbf{V}' \cup r, \mathbf{A}')$ be a compact (r, k) -core in D . Observe that every vertex in V' can be reached from r by a directed path all of whose internal vertices are in S_D and whose length is at most $2k - 1$. As every vertex in S_D has at most $20k^2$ out-neighbors, we obtain that the number of vertices that can be reached by such a path is at most $1 + 20k^2 + (20k^2)^2 + \dots + (20k^2)^{2k-1} \leq (20k^2)^{2k}$. As V' contains at most $2k - 1$ vertices, there are at most $\binom{(20k^2)^{2k}}{2k-1} \leq (20k^2)^{4k^2}$ possibilities to choose V' . Now suppose that we have chosen V' of size at most $2k - 1$. As there are at most four arcs in the same direction between any two vertices, there are at most $8 \binom{|V'|}{2} \leq 32k^2$ arcs that have their head and tail in $V' \cup r$. As all arcs of A' have both ends in $V' \cup r$, there are at most 2^{32k^2} possibilities to choose A' . It follows that there are at most $(20k^2)^{4k^2} \cdot 2^{32k^2}$ possibilities to choose a compact (r, k) -core X . As these can be computed by a brute force method, the algorithm can finish after checking less than $f(k) = \binom{(20k^2)^{4k^2} \cdot 2^{32k^2}}{2} = 2^{O(k^2 \cdot \log k)}$ candidates for the extendable pair of compact (r, k) -cores.

If no such extendable pair of arc-disjoint compact (r, k) -cores exists, by Lemma 4.4.11, D does not contain two arc-disjoint spanning (r, k) -flow branchings. On the other hand, if we find an extendable pair of arc-disjoint compact extendable (r, k) -cores, we also find such a pair (X_1, X_2) where X_1 and X_2 are arc-minimal, so by Lemma 4.4.6 triple-free. By the second part of Lemma 4.4.11, we can compute the two arc-disjoint spanning (r, k) -flow branchings in polynomial time. The overall running time of the obtained algorithm is $2^{O(k^2 \cdot \log k)} \cdot n^{O(1)}$. \square

4.4.4 An FPT algorithm for packing (r, k) -safe spanning trees

This section is concerned with proving Theorem 4.4.11. Again, slightly modifying the terminology introduced in Section 4.4.2 and reused in Section 4.4.3, given a singularly rooted graph $G = (V \cup r, E)$ and a positive integer k , we say that a vertex $v \in V \cup r$ is *large* if $|N_G(v)| \geq 8k - 7$, and *small* otherwise. And again, we let \mathbf{L}_G (resp. \mathbf{S}_G) be the set of vertices in V which are large (resp. small) in G .

We are now ready to introduce a new notion of certificates for (r, k) -safe spanning trees. A *compact (r, k) -certificate* is a subgraph $X = (V' \cup r, E')$ of G with $|V'| \leq 2k - 2$ satisfying the following:

- X is a tree,
- all vertices in $(V' \cup r) \cap L_G$ are leaves of X and
- an (r, k) -safe spanning tree Y can be obtained from X by adding a set V^* of $2k - 2 - |V'|$ new vertices and adding an edge from a vertex in $(V' \cup r) \cap L_G$ to v for all $v \in V^*$.

Observe that Y is not necessarily a subgraph of G . The following result, which is similar to Lemma 4.4.10 and Lemma 4.4.11, shows that compact certificates can be used in a similar way as classic certificates.

Lemma 4.4.12. *Let $G = (V \cup r, E)$ be a singularly rooted graph with $|V| \geq 2k - 2$. Then G has two edge-disjoint (r, k) -safe spanning trees if and only if G contains a completable pair of compact (r, k) -certificates. Further, given a completable pair of compact (r, k) -certificates, we can find a pair of edge-disjoint (r, k) -safe spanning trees in polynomial time.*

Proof. By Lemma 4.4.9, for the first part it suffices to prove that G contains a completable pair of compact (r, k) -certificates if and only if G contains a pair of completable classic (r, k) -certificates.

First let $(\mathbf{X}_1 = (\mathbf{V}_1 \cup r, \mathbf{E}_1), \mathbf{X}_2 = (\mathbf{V}_2 \cup r, \mathbf{E}_2))$ be a completable pair of classic (r, k) -certificates. Let $\mathbf{X}'_i = (\mathbf{V}'_i \cup r, \mathbf{E}'_i)$ be obtained from X_i by deleting $C_{X_i}^v$ for all $v \in (V_i \cup r) \cap L_G$. By construction, the X'_i are trees and all vertices in $(V'_i \cup r) \cap L_G$ are leaves in X'_i . Let \mathbf{Y}_i be obtained from X'_i by adding the vertices in $V_i - V'_i$ and adding an edge from a vertex $v \in (V'_i \cup r) \cap L_G$ to a vertex $w \in V_i - V'_i$ whenever $w \in V(C_{X_i}^v)$. Observe that Y_i is a tree with $|V(Y_i)| = 2k - 1$. Further, we have $|V_i - V(C_{Y_i}^v)| = |V_i - V(C_{X_i}^v)| \geq k$ for all $v \in V'_i$ and $|V(C_{Y_i}^v)| = 0$ for all $v \in V_i - V'_i$. This yields that $|V_i - V(C_{Y_i}^v)| \geq |V_i - V(C_{X_i}^v)| \geq k$ for all $v \in V_i$ and so Y_i is an (r, k) -safe tree. By definition, we obtain that (X'_1, X'_2) is a pair of compact (r, k) -certificates. Further, as $E(X'_i) \subseteq E(X_i)$ and (X_1, X_2) is completable, we obtain that (X'_1, X'_2) is completable.

Now let $(\mathbf{X}_1 = (\mathbf{V}_1 \cup r, \mathbf{E}_1), \mathbf{X}_2 = (\mathbf{V}_2 \cup r, \mathbf{E}_2))$ be a completable pair of compact (r, k) -certificates. By definition, there are (r, k) -safe trees $\mathbf{Y}_1, \mathbf{Y}_2$ such that Y_i is obtained from X_i by adding a set V_i^* of $2k - 2 - |V_i|$ new vertices and an edge from a vertex in $(V_i \cup r) \cap L_G$ to v for all $v \in V_i^*$. Let $(\mathbf{X}'_1 = (\mathbf{V}'_1 \cup r, \mathbf{E}'_1), \mathbf{X}'_2 = (\mathbf{V}'_2 \cup r, \mathbf{E}'_2))$ be a pair of subgraphs of G that are vertex-maximal with the following properties:

- (i) X'_i is obtained from X_i by repeatedly adding another vertex $v \in V - V_i$ and an edge of E that goes from a vertex in $(V_i \cup r) \cap L_G$ to v ,
- (ii) $d_{X'_i}(v) \leq d_{Y_i}(v)$ for all $v \in (V_i \cup r) \cap L_G$, and
- (iii) (X'_1, X'_2) is completable.

Note that (X_1, X_2) satisfies conditions (i)-(iii), so (X'_1, X'_2) is well-defined. Observe that if condition (ii) is satisfied with equality for all $v \in (V_i \cup r) \cap L_G$, then X'_i is isomorphic to Y_i , so X'_i is an (r, k) -safe tree and by definition also a classic (r, k) -certificate. If this is the case for both X'_1 and X'_2 , we are done by condition (iii).

We may therefore suppose by symmetry that there is a vertex $\mathbf{v} \in (V_1 \cup r) \cap L_G$ with $d_{X'_1}(\mathbf{v}) < d_{Y_1}(\mathbf{v})$. For any $\mathbf{e} = \mathbf{v}z_e \in \delta_G(\mathbf{v})$ with $z_e \in N_G(\mathbf{v}) - V'_1$, let $\mathbf{X}_e = (V'_1 \cup z_e, E'_1 \cup \mathbf{e})$. By the maximality of X'_1 , we obtain that X_e violates one of conditions (i)-(iii) for all

$e = vz_e \in \delta_G(v)$ with $z_e \in N_G(v) - V'_1$. By construction and the choice of v , X_e satisfies conditions **(i)** and **(ii)** for all $e = vz_e \in \delta_G(v)$ with $z_e \in N_G(v) - V'_1$. It follows that (X_e, X'_2) violates condition **(iii)** for all $e = vz_e \in \delta_G(v)$ with $z_e \in N_G(v) - V'_1$. As (X'_1, X'_2) is completable, there are two disjoint spanning trees T_1, T_2 of G such that $E'_i \subseteq E(T_i)$ for $i = 1, 2$.

Claim 4.4.1. *There is no $e = vz_e \in \delta_G(v) - (E(T_1) \cup E(T_2))$ with $z_e \in N_G(v) - V'_1$ such that (X_e, X'_2) violates condition **(iii)**.*

Proof. Suppose otherwise. By Lemma 4.4.2, there is an edge $f \in E(T_1)$ incident to z_e such that $T'_1 = (T_1 - f) \cup e$ is a spanning tree of G . As $z_e \notin V'_1$, we obtain that $f \notin E'_1$, yielding $E(X_e) \subseteq E(T'_1)$. As T'_1 and T_2 are edge-disjoint, we obtain that (X_e, X'_2) is completable, a contradiction. \square

Claim 4.4.2. *There are at most $6k - 6$ vertices $z \in N_G(v) - V'_1$ such that (X_e, X'_2) violates condition **(iii)** for some $e = vz \in E(T_2)$.*

Proof. Suppose otherwise. As $|V'_2| \leq |V(Y_2)| - 1 = 2k - 2$, we obtain that there are at least $4k - 3$ vertices $z \in N_G(v) - V'_1$ such that (X_e, X'_2) violates condition **(iii)** for some $e = vz \in E(T_2) - E_2$. Let $\sigma : E(T_2) \rightarrow E(T_1)$ be a tree-mapping function from T_2 to T_1 . By Lemma 2.1.1 and since $|E'_1| \leq 2k - 2$, there is some $z \in N_G(v) - V'_1$ and an edge $e = vz \in E(T_2) - E_2$ such that $\sigma(e) \in E(T_1) - E_1$. By definition of tree-mapping functions, $T'_1 = (T_1 - \sigma(e)) \cup e$ and $T'_2 = (T_2 - e) \cup \sigma(e)$ are edge-disjoint spanning trees of G . As $E(X_e) \subseteq E(T'_1)$ and $E(X_2) \subseteq E(T'_2)$, we obtain that (X_e, X'_2) is completable, a contradiction. \square

As $v \in L_G$ and $|V'_1| \leq |V_1 \cup V_1^*| = 2k - 2$, we have $|N_G(v) - V'_1| \geq (8k - 7) - (2k - 2) > 6k - 6$. It now follows from Claims 4.4.1 and 4.4.2 that there is at least one vertex in $z \in N_G(v) - V'_1$ and an edge $e = vz$ such that X_e does not violate any of conditions **(i)**-**(iii)**, a contradiction.

Observe that the second part of the proof yields an algorithm for computing a completable pair of classic (r, k) -certificates from (X_1, X_2) . Every time we try to add an edge e to X'_i , we test if (X_e, X'_{3-i}) satisfies conditions **(i)**-**(iii)**. Conditions **(i)**-**(ii)** can clearly be checked in polynomial time and, by Lemma 4.4.8, condition **(iii)** can also be checked in polynomial time. Never checking an edge that is parallel to one we have already checked, after at most $6k - 6$ failed attempts, we manage to add a new vertex to V'_i . We repeat this procedure $|V_i^*| \leq 2k - 2$ times. It follows that a completable pair of classic (r, k) -certificates can be computed in time $k^2 \cdot n^{O(1)} = n^{O(1)}$. By the second part of Lemma 4.4.9, we can then find two edge-disjoint (r, k) -safe spanning trees in G in polynomial time. The overall running time of the algorithm is polynomial, as claimed. \square

We are now ready to proceed to the proof of Theorem 4.4.11.

Proof. (of Theorem 4.4.11) We may suppose that there are at most two parallel edges from u to v for any $u, v \in V \cup r$. If $|V| < 2k - 2$, the problem can be solved by a brute force algorithm in time $2^{O(k^2)}$, by generating all pairs of subgraphs of G and checking whether any of these pairs satisfies the required conditions. We may hence also suppose that $|V| \geq 2k - 2$.

Let $\mathbf{X}_1 = (V_1 \cup r, E_1)$, $\mathbf{X}_2 = (V_2 \cup r, E_2)$ be two subgraphs of G . In order to test whether X_i is a compact (r, k) -certificate, we first check whether X_i is a tree such that all the vertices in $(V_i \cup r) \cap L_G$ are leaves of X . If this is the case, we add a set V_i^* of $2k - 2 - |V_i|$ new vertices to X_i . We then test all possibilities to add one edge from $(V_i \cup r) \cap L_G$ to v for all $v \in V_i^*$. As $|V_i \cup r| \leq 2k$ and $|V_i^*| \leq 2k$, there are at most $2k^{2k} = 2^{O(k \cdot \log k)}$ possibilities to check. For each of them, we can check in time

polynomial in k if the obtained graph is an (r, k) -safe spanning tree. By the definition of compact (r, k) -certificates, we can therefore check in time $2^{O(k \cdot \log k)}$ whether both X_1 and X_2 are compact (r, k) -certificates. By Lemma 4.4.8, we can therefore decide in time $2^{O(k \cdot \log k)} + n^{O(1)}$ if (X_1, X_2) is a completable pair of compact (r, k) -certificates in G . By Lemma 4.4.12, it therefore suffices to prove that there are at most $2^{O(k^2 \cdot \log k)}$ possible candidates for the completable pair of compact (r, k) -certificates, and that they can be generated within the same running time.

Let $\mathbf{X} = (\mathbf{V}' \cup r, \mathbf{E}')$ be a compact (r, k) -certificate in G . Observe that every vertex in V' can be reached from r by a path all of whose interior vertices are in S_G and whose length is at most $k - 1$. As every vertex in S_G has at most $8k - 8$ neighbors, we obtain that the number of vertices that can be reached by such a path is at most $1 + (8k - 8) + (8k - 8)^2 + \dots + (8k - 8)^{k-1} \leq (8k)^k$. As V' contains at most $2k - 2$ vertices, there are at most $\binom{(8k)^k}{2k-2} \leq (8k)^{2k^2}$ possibilities to choose V' .

Now suppose that we have chosen V' of size $2k - 2$. Observe that there are at most $2^{\binom{|V' \cup r|}{2}} \leq 8k^2$ edges that have both ends in $V' \cup r$. As $|E'| = 2k - 2$, there are at most $\binom{8k^2}{2k-2} \leq (8k^2)^{2k}$ possibilities to choose A' . It follows that there are at most $(8k)^{2k^2} \cdot (8k^2)^{2k}$ possibilities to choose a compact (r, k) -certificate X . As these can be computed by a brute force method, the algorithm can finish after checking less than $f(k) = \binom{(8k)^{2k^2} \cdot (8k^2)^{2k}}{2} = 2^{O(k^2 \cdot \log k)}$ candidates for the pair of compact (r, k) -certificates.

If no completable pair of compact (r, k) -certificates exists, by Lemma 4.4.9, G does not contain two edge-disjoint (r, k) -safe spanning trees. On the other hand, once we have found a pair of completable compact (r, k) -certificates, we can compute in polynomial time the two edge-disjoint (r, k) -safe spanning trees by the second part of Lemma 4.4.9. The overall running time of the obtained algorithm is $2^{O(k^2 \cdot \log k)} \cdot n^{O(1)}$. \square

4.4.5 A hardness result for packing (r, k) -safe spanning trees

In this section we prove Theorem 4.4.12. As pointed out in Section 2.3.3.1, the 3-SAT problem is NP-complete.

The proof of Theorem 4.4.12 given below is strongly inspired from the reduction given in [6, Theorem 5.2], but we provide it here entirely for the sake of completeness.

Proof. (of Theorem 4.4.12) Observe that, given a singularly rooted graph $G = (V \cup r, E)$ and two positive integers p and k , G contains an (r, k) -safe spanning tree if and only if the graph that is obtained from G by replacing each of its edges by p parallel copies of itself contains p edge-disjoint (r, k) -safe spanning trees. Hence, it suffices to prove the statement for $p = 1$. Let \mathbf{F} be an instance of 3-SAT, with variables $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_\ell$ and clauses $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_m$. Adding a variable that is not contained in any clause if necessary, we can assume that ℓ is even. We construct a singularly rooted graph $\mathbf{G} = (\mathbf{V} \cup r, \mathbf{E})$ as follows; see Figure 4.4.1 for an illustration.

For $i = 1, \dots, \ell$ let \mathbf{V}_i be an independent set containing two vertices \mathbf{v}_i and $\bar{\mathbf{v}}_i$, and let \mathbf{r} and \mathbf{t} be two extra vertices. Add all possible edges between r and V_1 , between V_i and V_{i+1} for $i = 1, \dots, \ell - 1$, and between V_ℓ and t . Next, add m vertices $\mathbf{c}_1, \dots, \mathbf{c}_m$ and link c_i to v_j (resp. \bar{v}_j) with a path containing $\ell/2$ interior vertices if x_i (resp. \bar{x}_i) is a literal of C_i . Finally, let $\mathbf{k} = 1 + \ell + 3\ell m/2 + m$ and add to G an independent set \mathbf{Q} on q vertices all linked to t , where $q > k - \ell - 1$ will be specified later. Notice that we have $n = |\mathbf{V}| = q + k + \ell + 1$.

We now prove that F is satisfiable if and only if G admits an (r, k) -safe spanning tree. First assume that G contains an (r, k) -safe spanning tree \mathbf{T} . Observe that, by construction and the definition of (r, k) -safe spanning tree, $\mathbf{T} - r$ has exactly two components whose vertex sets we denote by \mathbf{T}_1 and \mathbf{T}_2 . We may assume that \mathbf{T}_1 contains t . Then $G[\mathbf{T}_1]$

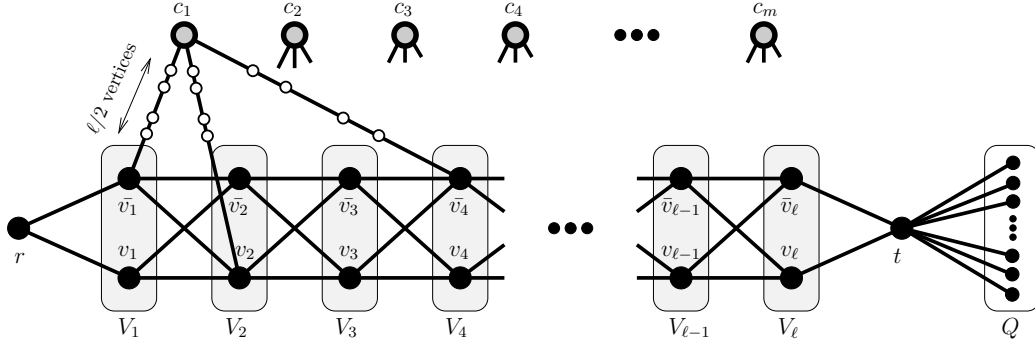


Figure 4.4.1: The singularly rooted graph G in the proof of Theorem 4.4.12 with $C_1 = \bar{x}_1 \vee x_2 \vee \bar{x}_4$.

contains a path \mathbf{P} from V_1 to t and T_1 contains Q . As a shortest path from V_1 to t contains $\ell + 1$ vertices, we have $|T_1| \geq q + \ell + 1$ and $n - |T_1| \leq k$. As T is (r, k) -safe, we obtain that $|T_1| = q + \ell + 1$ and T_1 consists exactly of the vertex set of the path P , which contains exactly $\ell + 1$ vertices, and the whole set Q . In particular, P intersects each V_i in exactly one vertex. Now for every variable x_i , we set x_i to TRUE if P contains \bar{v}_i and to FALSE if P contains v_i . For each clause C_j , there must be a path from the corresponding vertex c_j to V_1 in T_2 . Then one of the vertices corresponding to a literal of C_j must not be contained in P and so, this literal is set to TRUE and C_j is satisfied by it. It follows that the constructed assignment satisfies F .

Conversely, assume that F admits a truth assignment. Let \mathbf{P} be the path defined so that, for every $i = 1, \dots, \ell$, it contains v_i if x_i is set to FALSE and \bar{v}_i if x_i is set to TRUE. Further, let $\mathbf{T}_1 = V(\mathbf{P}) \cup t \cup Q$. As F is satisfied, for every $1 \leq j \leq m$, there exists a path from c_j to $V_1 \cup \dots \cup V_\ell$ in $G - (\mathbf{T}_1 \cup r)$. It follows that $G - (\mathbf{T}_1 \cup r)$ is connected and we select a spanning tree of it. The union of this spanning tree with $G[\mathbf{T}_1]$, r , and the edges incident to r is a spanning tree of G . In order to see that this spanning tree is (r, k) -safe, it suffices to observe that, as $q > k - \ell - 1$, we have $|\mathbf{T}_1| = q + \ell + 1 > k$ and $|V - \mathbf{T}_1| = q + k + 1 + \ell - (q + \ell + 1) = k$.

If we fix $q = k$, then the size of G is bounded by a polynomial in ℓ and m . Thus, the above reduction implies that, as 3-SAT is NP-complete, given a rooted graph G and an integer k , deciding whether G admits an (r, k) -safe spanning tree is NP-complete.

Let now ε be a positive constant and assume that k is an integer function satisfying $(\log(n))^{2+\varepsilon} \leq k(n) \leq \frac{n}{2}$ for all $n > 0$. Furthermore, suppose that there exists a constant C^* such that for all $c \geq C^*$ there exists an n such that $k(n) = c$. Finally, for the sake of a contradiction assume that there exists a polynomial-time algorithm \mathbf{A} , running in time $O(n^{c_0})$ for some $c_0 > 0$, for deciding if a given singularly rooted graph $G = (V \cup r, E)$ on n vertices contains an $(r, k(n))$ -safe spanning tree. Then let F be a 3-SAT formula with ℓ variables and m clauses. We may assume that ℓ and m are large enough so that $1 + \ell + 3\ell m/2 + m \geq C^*$. Adding copies of clauses which are already contained in the formula if necessary, we may also assume that $\ell \leq m$. By hypothesis, there exists n such that $k(n) = 1 + \ell + 3\ell m/2 + m$. So, in the above reduction, we choose q to be $n - (k(n) + \ell + 1)$, in order to have $n = q + k(n) + \ell + 1$. Then, using algorithm \mathbf{A} , one could decide if F is satisfiable in time $O(n^{c_0}) = O(2^{c_0 \cdot \log n}) = O(2^{c_0 \cdot k(n)^{1/(2+\varepsilon)}})$, where we have used the hypothesis that $k(n) \geq (\log(n))^{2+\varepsilon}$. Moreover, in the previous construction, we have $k(n) = 1 + \ell + 3\ell m/2 + m \leq 3m + 3m^2/2 \leq 3m^2$. So we could decide whether F is satisfiable in time $O(2^{c_0 \cdot (3m^2)^{1/(2+\varepsilon)}}) = O(2^{c'_0 \cdot m^{\varepsilon'}})$ with $\varepsilon' = 2/(2 + \varepsilon) < 1$, a contradiction to Lemma 2.3.3 assuming the ETH. \square

4.4.6 Conclusion

We considered three problems on finding certain disjoint substructures in graphs and digraphs. While in our proofs we restrict to finding two of these substructures for the sake of simplicity, our results can be generalized to allow for finding an arbitrary number of them using the same proof techniques. More concretely, the following results can be established using the techniques of this article. We omit the proofs of these generalized statements.

Theorem 4.4.13. *Given a singularly rooted digraph $D = (V \cup r, A)$ and an integer $p \geq 2$, deciding whether D contains p arc-disjoint k -safe spanning r -arborescences is FPT with parameter k . More precisely, the problem can be solved in time $2^{O(p \cdot k^2 \cdot \log k)} \cdot n^c$, where c is a constant depending on p . Further, if they exist, the p arc-disjoint k -safe spanning r -arborescences can be computed within the same running time.*

Theorem 4.4.14. *Given a singularly rooted digraph $D = (V \cup r, A)$ and an integer $p \geq 2$, deciding whether D contains p arc-disjoint (r, k) -flow branchings is FPT with parameter k . More precisely, the problem can be solved in time $2^{O(p \cdot k^2 \cdot \log k)} \cdot n^c$, where c is a constant depending on p . Further, if they exist, the p arc-disjoint (r, k) -flow branchings can be computed within the same running time.*

Theorem 4.4.15. *Given a singularly rooted graph $G = (V \cup r, E)$ and an integer $p \geq 2$, deciding whether G contains p arc-disjoint (r, k) -safe spanning trees is FPT with parameter k . More precisely, the problem can be solved in time $2^{O(p \cdot k^2 \cdot \log k)} \cdot n^c$, where c is a constant depending on p . Further, if they exist, the p edge-disjoint (r, k) -safe spanning trees can be computed within the same running time.*

It is natural to ask whether the dependency on k of our FPT algorithms can be improved. In the case of k -safe spanning r -arborescences (cf. Theorem 4.4.4), we can derive a lower bound from Theorem 4.4.2. Indeed, a corollary of Theorem 4.4.2 is that, assuming the ETH, for any two constants $\varepsilon > 0$ and $c > 0$, deciding whether a singularly rooted digraph contains two arc-disjoint k -safe spanning arborescences cannot be solved in time $2^{c \cdot k^{1-\varepsilon}} \cdot n^{O(1)}$. To see this, note that if such an algorithm existed, letting $k(n) := (\log(n))^{1+\varepsilon}$ we would obtain an algorithm in time $2^{c \cdot (\log(n))^{(1+\varepsilon)(1-\varepsilon)}} \cdot n^{O(1)} = n^{O(1)}$, contradicting Theorem 4.4.2. In other words, assuming the ETH, the problem cannot be solved in time $2^{O(k^{1-\varepsilon})} \cdot n^{O(1)}$ for any $\varepsilon > 0$.

Similarly to Theorem 4.4.2, Theorem 4.4.7 implies a lower bound for packing (r, k) -flow branchings (cf. Theorem 4.4.4): assuming the ETH, deciding whether a rooted digraph contains two arc-disjoint (r, k) -flow branchings cannot be solved in time $2^{O(k^{1-\varepsilon})} \cdot n^{O(1)}$ for any $\varepsilon > 0$. Also, concerning (r, k) -safe spanning trees (cf. Theorem 4.4.11), a consequence of Theorem 4.4.12 is that, assuming the ETH, for every $p \geq 1$, deciding whether a singularly rooted graph contains p edge-disjoint (r, k) -safe spanning trees cannot be solved in time $2^{O(k^{1/2-\varepsilon})} \cdot n^{O(1)}$ for any $\varepsilon > 0$.

There is still a significant gap between the above lower bounds, which are $2^{O(k^{1-\varepsilon})}$ or $2^{O(k^{1/2-\varepsilon})}$, and the function $2^{O(k^2 \cdot \log k)}$ in our FPT algorithms.

We did not focus on optimizing the polynomial factors in n of our algorithms, and we leave it for further research. Further, we leave as an open question whether any of the considered problems admits a polynomial kernel parameterized by k . Finally, it would be interesting to find a Theorem on packing k -safe mixed arborescences in mixed graphs, hence generalizing both Theorem 4.4.13 and Theorem 4.4.15.

Chapter 5

Connectivity augmentation

The problems we deal with in this section are augmentation problems. Given an undirected or directed graph, we wish to determine the minimum number of edges or arcs we need to add to the graph in order to obtain a certain connectivity property. We are interested in both minmax theorems for the necessary number of arcs or edges and in polynomial time algorithms for finding the actual augmentations.

For all problems treated in this section, the version with edge capacities and the version without edge capacities can be easily reduced to each other by replacing an edge with a capacity by multiple edges and vice-versa. Yet, this does not mean algorithmic equivalence. All the running times we give hold for the case with edge capacities, assuming that all basic operations can be executed in constant time. For this reason, all technical statements in Section 5.2 will be given in the capacitated form. In Section 5.1, however, we describe the problems in the uncapacitated form for the sake of simplicity.

In Section 5.1, we give a survey on the results and methods for the most basic connectivity augmentation problems, namely edge-, arc- and vertex-connectivity augmentation. In Section 5.2, we deal with the more specific problem of edge augmentation for $(2, k)$ -connectivity in undirected graphs.

5.1 Edge-, arc- and vertex-connectivity augmentation

We first deal with the problem on connectivity augmentation which is best understood, namely edge connectivity augmentation for undirected graphs. The following theorem that gives a minmax-formula for the number of edges needed to make a graph k -edge-connected was proven by Cai and Sun in [18].

Theorem 5.1.1. *Let $G = (V, E)$ be a graph and $k \geq 2$ an integer. Then the minimum number γ of edges needed to be added to G to obtain a k -edge-connected graph is*

$$\gamma = \left\lceil \frac{\max\{\sum_{X \in \mathcal{P}} (k - d_G(X)) : \mathcal{P} \text{ a subpartition of } V\}}{2} \right\rceil.$$

An algorithm that computes an actual augmentation and runs in polynomial time when k is fixed was provided by Watanabe and Nakamura in [110].

A different method of handling this problem was introduced by Frank [40]. This method allowed a simpler proof of Theorem 5.1.1 as well as an algorithm for computing the augmentation that runs in polynomial time even if k is part of the input. This method consists of two steps. In the first one a new vertex as well as edges connecting it to the given graph are added to have the required connectivity condition. Such a graph with

the minimum even number of new edges is called a *minimal even extension*. The second step applies splitting off. A minimum augmentation of G can be obtained by repeatedly applying splitting offs maintaining the connectivity requirements and finally deleting the added vertex. Using this method, several other augmentation problems can be treated similarly.

More precisely, Frank [40] provided an $O(n^5)$ time algorithm based on the approach described above. In order to achieve this running time, he uses a slightly involved method for the splitting off part. Carefully choosing the pairs of edges to be split off, he manages to finish with a complete splitting off after a linear number of splitting off operations in n while the obvious approach results in a quadratic number. Nagamochi and Ibaraki [94] provided an even more efficient implementation of the method described above. They managed to find a minimum k -edge-connected augmentation in $O((n \log n)(m + n \log n))$ time. An important ingredient in their work is an $O(n(m + n \log n))$ time mincut algorithm, see Lemma 5.2.1. We also make use of this mincut algorithm as a subroutine in Section 5.2.

Several generalizations of this concept have been considered. In [40], Frank proved a result that gives a minmax-formula for the number of edges needed to be added to a graph to make it satisfy some arbitrary given local edge-connectivity condition. We omit it here because it is slightly technical, but we give the following algorithmic corollary that is an immediate consequence of the proof of the theorem.

Theorem 5.1.2. *Let $G = (V, E)$ be a graph and let $r : V^2 \rightarrow \mathbb{Z}$ be a requirement function. Then a set E' of minimum cardinality such that for $G' = (V, E \cup E')$, we have $\lambda_{G'}(u, v) \geq r(u, v)$ can be found in $O(n^6)$.*

The method applied in the algorithm is the same as for the case of global edge-connectivity except that no way to linearize the number of splitting offs is known. Unfortunately, when we replace the minimum cardinality condition by a minimum cost condition with respect to an arbitrary cost function, there is no hope for a polynomial time algorithm even for the case of global connectivity. More concretely, the following result was proven by Eswaran and Tarjan [36].

Theorem 5.1.3. *Let $G = (V, E)$ be a graph, $k \in \mathbb{Z}_{\geq 0}$, $c : V^2 \rightarrow \mathbb{R}$ a cost function and $M \in \mathbb{R}$ a constant. Then it is NP-complete to decide whether there exists a new edge set E' such that $(V, E \cup E')$ is k -edge-connected and $c(E') \leq M$.*

On the other hand, Frank [40] provided a polynomial time algorithm when the costs are restricted to vertex costs.

Theorem 5.1.4. *Let $G = (V, E)$ be a graph, $k \in \mathbb{Z}_{\geq 0}$, $c : V \rightarrow \mathbb{R}$ a cost function and $M \in \mathbb{R}$ a constant. Then there is a polynomial time algorithm to decide whether there exists a new edge set E' such that $(V, E \cup E')$ is k -edge-connected and $\sum_{v \in V} c(v)d_{E'}(v) \leq M$.*

We next deal with arc-connectivity augmentation problems in directed graphs. The following fundamental theorem has been proven by Frank [40].

Theorem 5.1.5. *Let $D = (V, A)$ be a digraph. Then the minimum number γ of arcs needed to be added to D to obtain a k -arc-connected digraph is*

$$\gamma = \lceil \frac{1}{2} \max\{\max\{\sum_{X \in \mathcal{P}} (k - d_D^-(X)), \sum_{X \in \mathcal{P}} (k - d_D^+(X))\} : \mathcal{P} \text{ a subpartition of } V\} \rceil.$$

The proof of Theorem 5.1.5 again is similar to the proof of Theorem 5.1.1 and also yields a polynomial time algorithm for finding the actual augmentation. On the other hand, Frank also proved that the corresponding problem for local connectivity is not tractable in directed graphs.

Theorem 5.1.6. *Let $D = (V, A)$ be a digraph and let $r : V^2 \rightarrow \mathbb{Z}$ be a requirement function. Then computing a set A' of minimum cardinality such that for $D' = (V, A \cup A')$, we have $\lambda_{D'}(u, v) \geq r(u, v)$ is an NP-complete problem.*

Frank also proves in [40] that the algorithmic tractability of mincost versions in directed graphs behaves similarly to the one in undirected graphs.

Theorem 5.1.7. *Let $D = (V, A)$ be a digraph, $k \in \mathbb{Z}_{\geq 0}$, $c : V^2 \rightarrow \mathbb{R}$ a cost function and $M \in \mathbb{R}$ a constant. Then it is NP-complete to decide whether there exists a new arc set A' such that $(V, A \cup A')$ is k -arc-connected and $c(A') \leq M$.*

Theorem 5.1.8. *Let $D = (V, A)$ be a digraph, $k \in \mathbb{Z}_{\geq 0}$, $c^+, c^- : V \rightarrow \mathbb{R}$ cost functions and $M \in \mathbb{R}$ a constant. Then there is a polynomial time algorithm to decide whether there exists a new arc set A' such that $(V, A \cup A')$ is k -arc-connected and $\sum_{v \in V} (c^+(v)d_{A'}^+(v) + c^-(v)d_{A'}^-(v)) \leq M$.*

We now turn our attention to vertex-connectivity. In undirected graphs, we consider the problem of adding a set E' of minimum cardinality to a given graph in order to make it k -vertex-connected for some given positive integer k . A rather technical minmax theorem as well as an algorithm that runs in polynomial time for constant k have been provided by Jackson and Jordán [69].

The following is one of the most important open problems on connectivity augmentation, see [34].

Research Problem 5.1.1. *For the problem of finding a minimum k -vertex-connected augmentation of a given graph G , provide an algorithm that runs in polynomial time even if k is part of the input.*

In directed graphs, a rather involved minmax formula for the number of edges that need to be added to make a given graph k -vertex-connected for some positive integer k has been provided by Frank and Jordán [47]. This article also includes the first use of bisets. Bisets will also play a decisive role in Section 5.2. While their original proof did not yield a combinatorial polynomial time algorithm for computing the actual augmentation, the same authors gave a more elementary algorithm that runs in polynomial time for constant k in [46]. Finally, in contrast to the undirected case, an algorithm that also runs in polynomial time if k is part of the input has been provided by Végő and Benczúr in [107].

5.2 Mixed connectivity augmentation

We here deal with a problem generalizing vertex- and edge-connectivity augmentation. This section is based on [64].

The concept of (ℓ, k) -connectivity has been introduced by Kaneko and Ota in [72]. Clearly, (ℓ, k) -connectivity generalizes both edge- and vertex-connectivity. Most problems on mixed connectivity augmentation are wide open. The most desirable result would be a positive answer to the following problem.

Research Problem 5.2.1. *Given integers ℓ, k and a graph $G = (V, E)$, can we compute a minimum (ℓ, k) -connected augmentation of G in polynomial time?*

Clearly, a positive answer to Research Problem 5.2.1 would imply a positive answer to Research Problem 5.1.1. One might therefore first wish to consider the following weaker version where ℓ is fixed.

Research Problem 5.2.2. *Let ℓ be a fixed integer. Given an integer k and a graph $G = (V, E)$, can we compute a minimum (ℓ, k) -connected augmentation of G in polynomial time?*

An even weaker version can be obtained when also fixing the value of k .

Research Problem 5.2.3. *Let ℓ, k be fixed integers. Given a graph $G = (V, E)$, can we compute a minimum (ℓ, k) -connected augmentation of G in polynomial time?*

Similar questions can be asked for directed connectivity.

The purpose of this section is to give a positive answer for a more restricted special case, namely we give an affirmative answer to Research Problem 5.2.2 for $\ell = 2$. We strongly rely on the work of Durand de Gevigney and Szigeti in [27]. They prove the following minmax-formula for edge augmentation for $(2, k)$ -connectivity.

Theorem 5.2.1. *Let $(H = (V, E), c)$ be a capacitated graph and k a positive integer. Then the minimum number γ of edges that need to be added to obtain a $(2, k)$ -connected graph is*

$$\left\lceil \frac{1}{2} \max_{\mathcal{X} \in \mathcal{X}} \left\{ \sum_{X \in \mathcal{X}} (2k - (k|w(X)| + c(\delta_G(X_I, V - X_O))) \right\} \right\rceil.$$

Their proof follows the method of Frank. The creation of a minimum even $(2, k)$ -connected extension can easily be turned into a polynomial time algorithm. The splitting off procedure uses a splitting off theorem of [27] that will be introduced in Section 5.2.1 and turning it into a polynomial time algorithm is a little more complicated. While a careful analysis of the proof in [27] yields a rather slow polynomial time algorithm, no explicit mention of such an algorithm is made. The aim of this section is to provide a more efficient such algorithm. In Section 5.2.1, we give some results from [27] we need and prove some basic preliminary results. In Section 5.2.2, we show how to efficiently compute a minimal $(2, k)$ -connected extension. Further, we prove that a seeming difficulty in the splitting off process does not actually occur. This yields a rather simple $O(n^4(m + n \log n))$ time algorithm for the augmentation problem. In Section 5.2.3, we show how to obtain an algorithm that runs in $O(n^3(m + n \log n))$ time by linearizing the number of splitting offs.

5.2.1 Basic definitions and previous results

5.2.1.1 Minimum cuts

Given a capacitated graph $(G = (V, E), c)$, the mincut problem consists of finding a set $\emptyset \neq S \subsetneq V$ that minimizes $c(\delta_G(S))$. This problem has been widely studied. Due to the specific nature of our application, we are interested in a slight variation of this problem: given a capacitated graph $(H = (V \cup s, E), c)$ with a distinguished vertex $s \notin V$, we want to find a set $\emptyset \neq S \subsetneq V$ that minimizes $c(\delta_H(S))$. In other words, we additionally require that no side of the cut consists of s only. We denote the capacity of such a minimum cut by $\lambda_{(H,c)}(\mathbf{V})$. We say that $(H = (V \cup s, E), c)$ is k -edge-connected in V for some positive integer k if $\lambda_{(H,c)}(\mathbf{V}) \geq k$. We strongly rely on the following algorithmic result which is due to Nagamochi and Ibaraki [94].

Lemma 5.2.1. *Given a capacitated graph $(H = (V \cup s, E), c)$, we can compute $\lambda_{(H,c)}(\mathbf{V})$ and a set $\emptyset \neq S \subsetneq V$ that minimizes $c(\delta_H(S))$ in $O(n(m + n \log n))$ time.*

5.2.1.2 Bisets

Given a capacitated graph $(G = (V, E), c)$ and a positive integer k , we define a function f on the bisets on V by $\mathbf{f}(\mathbf{X}) = k|w(\mathbf{X})| + c(\delta_G(X_I, V - X_O))$. Observe that $f(\mathbf{X}) = f(\overline{\mathbf{X}})$. This function will play a crucial role throughout this section.

We can now rephrase the definition of $(2, k)$ -connectivity in terms of bisets. A capacitated graph $(G = (V, E), c)$ with $|V| \geq 3$ is $(2, k)$ -connected if for every biset \mathbf{X} which is nontrivial with respect to V , we have $f(\mathbf{X}) \geq 2k$. We also need the following slightly more advanced notion: A capacitated graph $(H = (V \cup s, E), c)$ with $|V| \geq 3$ is called $(2, k)$ -connected in V if for every biset \mathbf{X} on V which is nontrivial with respect to V , we have $f(\mathbf{X}) \geq 2k$. Observe that in $H = (V \cup s, E)$ the vertex s belongs to $\overline{\mathbf{X}}$ for any $X \subseteq V$.

5.2.1.3 Splitting off

Let $(H = (V \cup s, E), c)$ be a capacitated graph. For $v \in N_H(s)$ and a nonnegative integer $\alpha \leq c(sv)$, we denote by $(\mathbf{H}, \mathbf{c})_v^\alpha$ the capacitated graph obtained from (H, c) by decreasing the capacity of sv by α . If $c(sv) = 0$ after the operation, we delete sv from H . For (H, c) that is $(2, k)$ -connected in V , we denote $\mathbf{U}_{(\mathbf{H}, \mathbf{c})} = \{v \in V \mid (H, c)_v^1 \text{ is } (2, k)\text{-connected in } V\}$, a set that will play a significant role later on. For a vertex $v \in N_H(s)$, we denote by $(\mathbf{H}, \mathbf{c})_v^{\max}$ the capacitated graph $(H, c)_v^\alpha$ where α is the maximum integer such that $(H, c)_v^\alpha$ is well-defined and $(2, k)$ -connected in V .

For $u, v \in N_H(s)$ and a positive integer $\alpha \leq \min\{c(su), c(sv)\}$, we denote by $(\mathbf{H}, \mathbf{c})_{u,v}^\alpha$ the capacitated graph obtained from (H, c) by decreasing $c(su)$ and $c(sv)$ by α and increasing $c(uv)$ by α . We delete edges of capacity 0 and create the edge uv if it does not exist yet. We also delete the arising loop if $u = v$. We call this operation the α -multiple splitting off of su and sv and say that this α -multiple splitting off contains su and sv . We abbreviate 1-multiple splitting off to *splitting off*. Suppose that (H, c) is $(2, k)$ -connected in V . We say that a pair (su, sv) is *admissible* if $(H, c)_{u,v}^1$ is $(2, k)$ -connected in V . For $u, v \in N_H(s)$, let α be the maximum integer such that $(H, c)_{u,v}^\alpha$ is well-defined and $(2, k)$ -connected in V . We call an α -multiple splitting off of (su, sv) a *maximal splitting off* of (su, sv) and denote $(\mathbf{H}, \mathbf{c})_{u,v}^{\max} = (H, c)_{u,v}^\alpha$. Observe that every maximal splitting off can be viewed as a series of splitting offs.

We next give an important characterization of admissible pairs in H . Given a pair (su, sv) , a biset \mathbf{X} which is nontrivial with respect to V with either $f(\mathbf{X}) \leq 2k + 1$ and $u, v \in X_I$ or $f(\mathbf{X}) = 2k, u, v \in X_O$ and $\{u, v\} \cap X_I \neq \emptyset$ is said to *block* (su, sv) . The following result can be found as Lemma 3.1 in [27] and will be frequently used.

Lemma 5.2.2. *Given a capacitated graph $(H = (V \cup s, E), c)$ that is $(2, k)$ -connected in V and $u, v \in N_H(s)$, (su, sv) is admissible if and only if there is no biset blocking it.*

A biset that blocks a pair of edges (su, sv) with $u \neq v$ is called *horrifying*. Note that the wall of a horrifying biset contains at most one vertex. Further, observe that we can check whether a given biset is horrifying in $O(m)$ time by applying the definition.

While the following result is not explicitly proven in [27], its proof is almost literally the same as the one of Lemma 3.4 in [27]. We therefore omit it. The result nevertheless plays a key role in our algorithm.

Lemma 5.2.3. *Let $(H = (V \cup s, E), c)$ be a capacitated graph that is $(2, k)$ -connected in V with $c(\delta_H(s))$ even. Let \mathbf{X} be a horrifying biset, $u \in X_I \cap N_H(s)$ and $v \in N_H(s) - X_I$. If a biset \mathbf{Y} blocks (su, sv) , then either $\mathbf{X} \cup \mathbf{Y}$ is horrifying or \mathbf{X} and \mathbf{Y} have the same wall of size 1.*

We also require the following result that can be found in [27] as Proposition 3.2.

Lemma 5.2.4. *Let $(H = (V \cup s, E), c)$ be a capacitated graph that is $(2, k)$ -connected in V and let X be a horrifying biset. Then $N_H(s) - X_O \neq \emptyset$.*

A series of splitting offs at s that results in a capacitated graph in which s is an isolated vertex is called a *complete splitting off* of (H, c) . It is easy to see that a complete splitting off exists if and only if $c(\delta_H(s))$ is even. A complete splitting off is *admissible* if each of the splitting offs it contains is admissible in the current capacitated graph when being chosen. This is equivalent to the finally obtained capacitated graph being $(2, k)$ -connected after deleting s . Finding such a complete admissible splitting off is the main difficulty in our algorithm. We strongly rely on a characterization of capacitated graphs having a complete admissible splitting off that can be found in [27]. Before stating it, we need the following definition:

Let $(H = (V \cup s, E), c)$ be a capacitated graph that is $(2, k)$ -connected in V and with $c(\delta_H(s))$ even. An *obstacle* is a collection \mathcal{B} of bisets in V and a vertex $t \in N_H(s)$ satisfying the following:

$$c(st) \text{ is odd and } t \in U_{(H,c)}, \quad (5.2.1)$$

$$w(\mathbf{B}) = \{t\} \text{ and } f(\mathbf{B}) = 2k \text{ for all } \mathbf{B} \in \mathcal{B}, \quad (5.2.2)$$

$$B_I \cap B'_I = \emptyset \text{ for all distinct } \mathbf{B}, \mathbf{B}' \in \mathcal{B}, \quad (5.2.3)$$

$$N_H(s) - \{t\} \subseteq \bigcup_{\mathbf{B} \in \mathcal{B}} B_I. \quad (5.2.4)$$

We say that t is the *special* vertex of the obstacle. It is easy to see that a capacitated graph containing an obstacle does not have a complete admissible splitting off, as every splitting off of (st, su) for some $u \in N_H(s)$ is blocked by the biset $\mathbf{B} \in \mathcal{B}$ with $u \in B_I$ and by Lemma 5.2.2. In [27], it is proved that the converse is also true:

Theorem 5.2.2. *Let $(H = (V \cup s, E), c)$ be a capacitated graph that is $(2, k)$ -connected in V for some $k \geq 2$ with $c(\delta_H(s))$ even. Then (H, c) has a complete admissible splitting off at s if and only if (H, c) does not contain an obstacle.*

5.2.1.4 Basic algorithms

In this section, we show that we can efficiently compute the maximum decrease of the capacity of an edge and the maximum multiplicity of a splitting off of an edge pair that maintain certain connectivity requirements. We first show this for the case of edge-connectivity and then apply this for the case of $(2, k)$ -connectivity. All of these results are simple consequences of Lemma 5.2.1.

Lemma 5.2.5. *Given a capacitated graph $(H = (V \cup s, E), c)$ that is k -edge-connected in V and a vertex $v \in N_H(s)$, we can compute the maximal α such that $(H', c') = (H, c)_v^\alpha$ is k -edge-connected in V in $O(n(m+n \log n))$ time. Further, if $c'(sv) \neq 0$, we can compute a set S with $v \in S \subsetneq V$ and $c'(\delta_{H'}(S)) = k$ in $O(n(m+n \log n))$ time.*

Proof. Let $\gamma = c(sv)$ and $(H_\gamma, c_\gamma) = (H, c)_v^\gamma$. Obviously we have $\alpha \leq \gamma$. The other condition α needs to satisfy is that $c'(\delta_{H'}(X)) \geq k$ for every $\emptyset \neq X \subsetneq V$. If $v \notin X$, we obtain $c'(\delta_{H'}(X)) = c(\delta_H(X)) \geq k$. If $v \in X$, the condition is satisfied if and only if $0 \geq k - c'(\delta_{H'}(X)) = k - (c_\gamma(\delta_{H_\gamma}(X)) - \alpha + \gamma) = \alpha - (\gamma - k + c_\gamma(\delta_{H_\gamma}(X)))$. It follows that $\alpha = \min\{\gamma, \gamma - k + \lambda_{(H_\gamma, c_\gamma)}(V)\}$. By Lemma 5.2.1, we can compute $\lambda_{(H_\gamma, c_\gamma)}(V)$, and hence α , and also a set $S \subsetneq V$ with $c_\gamma(\delta_{H_\gamma}(S)) = \lambda_{(H_\gamma, c_\gamma)}(V)$ in $O(n(m+n \log n))$ time. If $c'(sv) \neq 0$, we have $v \in S$ and $c'(\delta_{H'}(S)) = k$. \square

Lemma 5.2.6. *Given a capacitated graph $(H = (V \cup s, E), c)$ that is k -edge-connected in V and vertices $u, v \in N_H(s)$, we can compute the maximal α such that $(H', c') = (H, c)_{u,v}^\alpha$ is k -edge-connected in V in $O(n(m + n \log n))$ time. Further, if $c'(su), c'(sv) \neq 0$, we can compute a set S with $u, v \in S \subsetneq V$ and $c'(\delta_{H'}(S)) \leq k + 1$ in $O(n(m + n \log n))$ time.*

Proof. Let $\gamma = \min\{c(su), c(sv)\}$ and $(H_\gamma, c_\gamma) = (H, c)_{u,v}^\gamma$. Obviously we have $\alpha \leq \gamma$. The other condition α needs to satisfy is that $c'(\delta_{H'}(X)) \geq k$ for every $\emptyset \neq X \subsetneq V$. If $\{u, v\} - X \neq \emptyset$, we obtain $c'(\delta_{H'}(X)) = c(\delta_H(X)) \geq k$. If $u, v \in X$, the condition is satisfied if and only if $0 \geq k - c'(\delta_{H'}(X)) = k - (c_\gamma(\delta_{H_\gamma}(X)) - 2\alpha + 2\gamma) = 2(\alpha - (\gamma + \frac{1}{2}c_\gamma(\delta_{H_\gamma}(X)) - \frac{1}{2}k))$. It follows that $\alpha = \min\{\gamma, \lfloor \gamma + \frac{1}{2}\lambda_{(H_\gamma, c_\gamma)}(V) - \frac{1}{2}k \rfloor\}$. By Lemma 5.2.1, we can compute $\lambda_{(H_\gamma, c_\gamma)}(V)$, and hence α , and also a set $S \subsetneq V$ with $c_\gamma(\delta_{H_\gamma}(S)) = \lambda_{(H_\gamma, c_\gamma)}(V)$, in $O(n(m + n \log n))$ time. If $c'(su), c'(sv) \neq 0$, we have $u, v \in S$ and $c'(\delta_{H'}(S)) \leq k + 1$. \square

Lemma 5.2.7. *Given a capacitated graph $(H = (V \cup s, E), c)$ that is $(2, k)$ -connected in V and a vertex $v \in N_H(s)$, we can compute $(H, c)_v^{max}$ in $O(n^2(m + n \log n))$ time.*

Proof. By definition, $(H, c)_v^{max} = (H, c)_v^\alpha$ where α is the maximum integer such that $(H, c)_v^\alpha$ is $2k$ -edge-connected in V and $(H, c)_v^\alpha - x$ is k -edge-connected in $V - x$ for all $x \in V$. We first compute the maximum integer α' such that $(H, c)_v^{\alpha'}$ is $2k$ -edge-connected in V . Using Lemma 5.2.5, this can be done in $O(n(m + n \log n))$ time. Next observe that for any nonnegative integer β , $(H, c)_v^\beta - v = (H, c) - v$ is always k -edge-connected in $V - v$ by assumption. Now consider $x \in V - v$ and observe that for any nonnegative integer β , we have $(H, c)_v^\beta - x = (H - x, c)_v^\beta$. It follows from Lemma 5.2.5 that we can compute the maximum integer α_x such that $(H, c)_v^{\alpha_x} - x$ is k -edge-connected in $V - x$ in $O(n(m + n \log n))$ time. We now can compute $\alpha = \min\{\alpha', \min_{x \in V - v} \alpha_x\}$. The overall running time is $O(n^2(m + n \log n))$. \square

Lemma 5.2.8. *Given a capacitated graph $(H = (V \cup s, E), c)$ that is $(2, k)$ -connected in V and vertices $u, v \in N_H(s)$, we can compute $(H', c') = (H, c)_{u,v}^{max}$ in $O(n^2(m + n \log n))$ time. Further, if $c'(su), c'(sv) \neq 0$, we can compute a biset blocking (su, sv) in (H', c') in $O(n^2(m + n \log n))$ time.*

Proof. By definition, $(H', c') = (H, c)_{u,v}^\alpha$ where α is the maximum integer such that $(H, c)_{u,v}^\alpha$ is $2k$ -edge-connected in V and $(H, c)_{u,v}^\alpha - x$ is k -edge-connected in $V - x$ for all $x \in V$. We first compute the maximum integer α' such that $(H, c)_{u,v}^{\alpha'}$ is $2k$ -edge-connected in V . Using Lemma 5.2.6, this can be done in $O(n(m + n \log n))$ time. For $x \in V - \{u, v\}$, we can compute, by Lemma 5.2.6, the maximum integer α_x such that $(H, c)_{u,v}^{\alpha_x} - x = (H - x, c)_{u,v}^{\alpha_x}$ is k -edge-connected in $V - x$ in $O(n(m + n \log n))$ time. We can compute, by Lemma 5.2.5, the maximum integer α_u such that $(H, c)_{u,v}^{\alpha_u} - u = (H - u, c)_v^{\alpha_u}$ is k -edge-connected in $V - u$ in $O(n(m + n \log n))$ time. We similarly compute α_v . We now can compute $\alpha = \min\{\alpha', \min_{x \in V} \alpha_x\}$. The overall running time is $O(n^2(m + n \log n))$.

Now suppose that $c'(su), c'(sv) \neq 0$. If $\alpha = \alpha'$, that is $(H', c') = (H, c)_{u,v}^{\alpha'}$, then, by Lemma 5.2.6, a set $S \subsetneq V$ with $u, v \in S$ and $c'(\delta_{H'}(S)) \leq 2k + 1$ can be found in $O(n(m + n \log n))$ time. We obtain that (S, S) is a biset blocking (su, sv) in (H', c') . If $\alpha = \alpha_x$ for some $x \in V - \{u, v\}$, that is $(H', c') - x = (H - x, c)_{u,v}^{\alpha_x}$, then, by Lemma 5.2.6, a set $S \subsetneq V - x$ with $u, v \in S$ and $c'(\delta_{H' - x}(S)) \leq k + 1$ can be found in $O(n(m + n \log n))$ time. We obtain that $(S \cup \{x\}, S)$ is a biset blocking (su, sv) in (H', c') . Finally, if $\alpha = \alpha_u$ or α_v , say α_u , that is $(H', c') - u = (H - u, c)_v^{\alpha_u}$, then, by Lemma 5.2.5, a set $S \subsetneq V - u$ with $v \in S$ and $c'(\delta_{H'}(S)) = k$ can be found in $O(n(m + n \log n))$ time. We obtain that $(S \cup \{u\}, S)$ is a biset blocking (su, sv) in (H', c') . \square

5.2.2 Minimal even extensions for $(2, k)$ -connectivity

A minimal even extension for $(2, k)$ -connectivity of a capacitated graph $(G = (V, E_0), c_0)$ is obtained by adding a new vertex and edges incident to this vertex so that the obtained capacitated graph becomes $(2, k)$ -connected in V and so that the total capacity of the new edges is even and minimal. The importance of minimal even extensions for $(2, k)$ -connectivity is due to a theorem from [27] that shows that minimum augmentations for $(2, k)$ -connectivity can be computed from minimal even extensions for $(2, k)$ -connectivity by a complete admissible splitting off. We first give a simple algorithm to compute minimal even extensions for $(2, k)$ -connectivity and give some basic properties. We further show a property of minimal even extensions for $(2, k)$ -connectivity which is essential to our splitting off algorithms. This then allows us to give a naive algorithm for the $(2, k)$ -connectivity augmentation problem which is slower than the algorithm which is the main result of this section and is given later.

We first introduce the algorithm for computing a minimal even extension for $(2, k)$ -connectivity. In order to avoid a technical definition whose details are not essential to this work, Algorithm 1 will also serve as a definition for minimal even extensions for $(2, k)$ -connectivity. Algorithm 1 takes a capacitated graph $(G = (V, E_0), c_0)$ as input and adds a new vertex s as well as edges of sufficiently high capacity between s and all other vertices to make the capacitated graph $(2, k)$ -connected in V . It then reduces these capacities in a greedy way as much as possible while maintaining $(2, k)$ -connectivity in V . Finally, if the degree of s is odd, it augments the capacity of a certain chosen edge by 1.

Algorithm 1 Minimal even extensions for $(2, k)$ -connectivity

Input: A capacitated graph $(G = (V, E_0), c_0)$, an integer $k \geq 2$.
Output: A minimal even extension for $(2, k)$ -connectivity of (G, c_0) .

- 1 Create (H, c) by adding a vertex s to V and adding an edge of capacity $2k$ between s and every $v \in V$;
- 2 Let (v_1, \dots, v_n) be an arbitrary ordering of the vertices of V ;
- 3 **for** $i = 1, \dots, n$ **do**
- 4 $(H, c) = (H, c)_{v_i}^{max}$;
- 5 **if** $c(\delta_H(s))$ is odd **then**
- 6 choose the maximum i^* such that $c(sv_{i^*})$ is odd;
- 7 $c(sv_{i^*}) = c(sv_{i^*}) + 1$;
- 8 **Return** (H, c) ;

Given a capacitated graph (G, c_0) , a capacitated graph (H, c) which is obtained by applying Algorithm 1 to (G, c_0) is called a *minimal even extension for $(2, k)$ -connectivity* of (G, c_0) .

Proposition 5.2.1. *Given a capacitated graph (G, c_0) , a minimal even extension for $(2, k)$ -connectivity of (G, c_0) can be computed in $O(n^3(m + n \log n))$ time.*

Proof. By definition, a minimal even extension for $(2, k)$ -connectivity can be computed by Algorithm 1. It follows from Lemma 5.2.7 that line 4 can be executed in $O(n^2(m + n \log n))$ time. As line 4 is executed n times and the rest of Algorithm 1 can be executed efficiently, Algorithm 1 runs in $O(n^3(m + n \log n))$ time. \square

We next collect some basic properties of minimal even extensions for $(2, k)$ -connectivity.

Proposition 5.2.2. *Let $(H = (V \cup s, E), c)$ be a minimal even extension for $(2, k)$ -connectivity of a capacitated graph $(G = (V, E_0), c_0)$. Then the following hold:*

- (a) (H, c) is $(2, k)$ -connected in V ,
- (b) $c(\delta_H(s))$ is even,
- (c) $c(sv)$ is even for all $v \in U_{(H, c)}$,
- (d) $(H, c)_v^2$ is not $(2, k)$ -connected in V for any $v \in V$.

Proof. We obtain (a) as an immediate consequence of the construction of Algorithm 1.

If the if-condition in line 5 is not satisfied, $c(\delta_H(s))$ is even and remains unchanged in the rest of the algorithm. Otherwise, $c(\delta_H(s))$ is odd and is augmented by 1 in line 7. This yields (b).

We denote by (H_i, c_i) the capacitated graph defined in line 4 in iteration i . Since line 4 is executed, for all $i \in \{1, \dots, n\}$ with $c_i(sv_i) \geq 1$, there exists a biset \mathbf{X}^i such that $v_i \in X_I^i$ and $f_{(H_i, c_i)}(\mathbf{X}^i) = 2k$. If the if-condition in line 5 is not satisfied, then $f_{(H, c)}(\mathbf{X}^i) \leq f_{(H^i, c^i)}(\mathbf{X}^i) = 2k$ for all $i \in \{1, \dots, n\}$ with $c_i(sv_i) \geq 1$ and so $U_{(H, c)} = \emptyset$. If the if-condition in line 5 is satisfied, then i^* is defined in line 6. For all $i \in \{1, \dots, i^* - 1\}$ with $c_i(sv_i) \geq 1$, since $c_i(sv_i) \geq 1$, $c_i(sv_{i^*}) = 2k$ and $f_{(H_i, c_i)}(\mathbf{X}^i) = 2k$, we have $v_{i^*} \notin X_I^i$. This yields $f_{(H, c)}(\mathbf{X}^i) \leq f_{(H^i, c^i)}(\mathbf{X}^i) = 2k$, so $v_i \notin U_{(H, c)}$. For all $i \in \{i^*, \dots, n\}$, $c(sv_i)$ is even after the execution of line 7. This proves (c).

For any $v_i \in V$ with $c(sv_i) \geq 1$, we have $f_{(H, c)}(\mathbf{X}^i) \leq f_{(H^i, c^i)}(\mathbf{X}^i) + 1 = 2k + 1$. This proves (d). \square

We now give the following theorem which is the reason for us considering minimal even extensions for $(2, k)$ -connectivity. Its proof can be found in [27]. It shows that minimum augmentations for $(2, k)$ -connectivity can be computed from minimal even extensions for $(2, k)$ -connectivity if we can find a complete admissible splitting off at s .

Theorem 5.2.3. *Let $(G = (V, E_0), c_0)$ be a capacitated graph, $(H = (V \cup s, E), c)$ a minimal even extension for $(2, k)$ -connectivity of (G, c_0) and let (H', c') be obtained from (H, c) by a complete admissible splitting off. Then $(H', c') - s$ is a minimum $(2, k)$ -connected augmentation of (G, c_0) .*

We now prove an important result that shows that when finding a complete admissible splitting off, we do not need to worry about obstacles.

Lemma 5.2.9. *Let $(H = (V \cup s, E), c)$ be a minimal even extension for $(2, k)$ -connectivity of a capacitated graph (G, c_0) and let (H_1, c_1) be obtained from (H, c) by a series of admissible splitting offs. Then (H_1, c_1) contains no obstacle.*

Proof. Suppose that (H_1, c_1) contains an obstacle \mathcal{B} with special vertex t . By (5.2.1), $c_1(st)$ is odd and $t \in U_{(H_1, c_1)} \subseteq U_{(H, c)}$. It follows by Proposition 5.2.2 (c) that $c(st)$ is even, so st was split off with an edge sv . By Proposition 5.2.2 (d), we have $v \neq t$. Let (H_2, c_2) be the capacitated graph such that $(H_1, c_1) = (H_2, c_2)_{t, v}^1$.

Claim 5.2.1. (H_2, c_2) contains no obstacle.

Proof. Suppose otherwise, so (H_2, c_2) contains an obstacle \mathcal{B}' . By (5.2.1), the special vertex of \mathcal{B}' cannot be t because $c_2(st) = c_1(st) + 1$ is even. It follows by (5.2.4) that $t \in B_I'$ for some $\mathbf{B}' \in \mathcal{B}'$. Now (5.2.2) yields $f_{(H_2, c_2)}(\mathbf{B}') = 2k$ which contradicts that by (5.2.1), $t \in U_{(H_1, c_1)} \subseteq U_{(H_2, c_2)}$. \square

By Claim 5.2.1 and Theorem 5.2.2, (H_2, c_2) has a complete admissible splitting off. In particular, as $c_2(st) \geq 2$, there exist $x, y \in N_{H_2}(s)$ (possibly $x = y$) such that $(H_3, c_3) = ((H_2, c_2)_{t,x}^1)_{t,y}^1$ is $(2, k)$ -connected in V . By Proposition 5.2.2(d), $(H, c)_{t,t}^1 = (H, c)_t^2$ is not $(2, k)$ -connected in V and hence neither is $(H_2, c_2)_{t,t}^1$, so $x, y \neq t$. Obviously $x, y \neq v$, so, by (5.2.4), $x, y \in \bigcup_{B \in \mathcal{B}} B_I$. Then, by (5.2.2), we have

$$\begin{aligned} 2 + |\mathcal{B}|k &\leq 2 + \sum_{B \in \mathcal{B}} c_3(\delta_{H_3-t}(B_I)) \\ &= \sum_{B \in \mathcal{B}} c_2(\delta_{H_2-t}(B_I)) \\ &= 1 + \sum_{B \in \mathcal{B}} c_1(\delta_{H_1-t}(B_I)) \\ &= 1 + |\mathcal{B}|k, \end{aligned}$$

a contradiction. □

We are now ready to give a first naive algorithm for finding a complete admissible splitting off of minimal even extensions for $(2, k)$ -connectivity.

Algorithm 2 Naive splitting off

Input: A minimal even extension for $(2, k)$ -connectivity $(H = (V \cup s, E), c)$ of a capacitated graph (G, c_0) .

Output: A minimum $(2, k)$ -connected augmentation of (G, c_0) .

- 1 **for** $u \neq v \in N_H(s)$ **do**
 - 2 $(H, c) = (H, c)_{u,v}^{max}$;
 - 3 **Return** $(H, c) - s$;
-

By Lemma 5.2.9, no obstacle is created during the execution of Algorithm 2 and so, by Theorems 5.2.2 and 5.2.3, the output of Algorithm 2 is a minimum $(2, k)$ -connected augmentation of (G, c_0) . As line 2 is executed at most n^2 times and by Lemma 5.2.8, Algorithm 2 runs in $O(n^4(m + n \log n))$ time. Together with Algorithm 1, this yields an $O(n^4(m + n \log n))$ time algorithm for the $(2, k)$ -connectivity augmentation problem.

5.2.3 A fast splitting off algorithm

This section is dedicated to refining Algorithm 2 in order to improve its running time from $O(n^4(m + n \log n))$ time to $O(n^3(m + n \log n))$ time. Together with Algorithm 1 and Theorem 5.2.3, this yields an $O(n^3(m + n \log n))$ time algorithm for the $(2, k)$ -connectivity augmentation problem.

While Algorithm 2 executes maximal splitting offs for all pairs of edges incident to s and therefore executes maximal splitting offs for a number of pairs which is quadratic in n before terminating, the refined version, Algorithm 3, carefully chooses the pairs to be split off. This allows us to terminate after a number of maximal splitting offs which is linear in n .

In order to achieve this, Algorithm 3 maintains not only a capacitated graph that is obtained from the minimal even extension for $(2, k)$ -connectivity that is its input by a series of splitting offs. It also stores the information obtained from the fact that certain pairs are not admissible in the form of a biset \mathbf{X} . If two edges incident to s both have their second endvertex in the inner set of \mathbf{X} , their splitting off is not admissible. The

maintenance of \mathbf{X} therefore allows us to avoid attempts of splitting offs of pairs which are known to be nonadmissible.

During each iteration of the algorithm, we execute one or two maximal splitting offs. If none of these maximal splitting offs delete an edge incident to s , we modify \mathbf{X} . The number of neighbors of s which are not covered by X_O never increases. Further, after a small constant number of iterations of our algorithm, either an edge incident to s is deleted or the number of neighbors of s not covered by X_O decreases. This allows us to obtain the desired running time.

In the first part of this section, we show a key lemma that is needed to modify \mathbf{X} in a favorable way. After, we describe the algorithm in the form of a pseudocode. Finally, we prove the correctness of the algorithm and analyze its running time.

5.2.3.1 Key lemma

This part gives a result that allows us to modify the biset \mathbf{X} .

Lemma 5.2.10. *Let $(H = (V \cup s, E), c)$ be a capacitated graph that is $(2, k)$ -connected in V for some $k \geq 2$ and has a complete admissible splitting off. Let \mathbf{X} be a horrifying biset, $u \in X_I \cap N_H(s), v \in N_H(s) - X_O, \mathbf{Y}$ a biset blocking $(su, sv), z \in (X_I - Y_I) \cap N_H(s)$ and suppose that $\mathbf{X} \cup \mathbf{Y}$ is not horrifying. Let \mathbf{Z} be a biset blocking (sv, sz) . Then $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$ is horrifying.*

Proof. By Lemma 5.2.3, we have $w(\mathbf{X}) = w(\mathbf{Y}) = \{p\}$ for some vertex $p \in V$. Since $\{u\}, \{v\} \neq w(\mathbf{X}) = w(\mathbf{Y})$ and \mathbf{Y} blocks (su, sv) , we have $u, v \in Y_I$. Let $(\mathbf{H}', \mathbf{c}') = (H, c) - p$ and $\mathbf{d}'(S) := c'(\delta_{H'}(S))$ for $S \subseteq V - p$. For $S_1, S_2 \subseteq V - p$, we use $\overline{\mathbf{d}'}(\mathbf{S}_1, \mathbf{S}_2)$ for $c'(\delta_{H'}(S_1 \cap S_2, ((V \cup s) - p) - (S_1 \cup S_2)))$. Observe that (H', c') is k -edge-connected in $V - p$ since (H, c) is $(2, k)$ -connected in V . This yields that $\mathbf{d}'(S) \geq k$ for any $\emptyset \neq S \subsetneq V - p$.

We distinguish two cases depending on where the wall of \mathbf{Z} is located.

Case 5.2.1. $w(\mathbf{Z}) = \{p\}$.

We show that in this case $\mathbf{X} \cup \mathbf{Y} \subseteq \mathbf{Z}$ and hence $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} = \mathbf{Z}$ is horrifying. For the sake of a contradiction, suppose that $(X_I \cup Y_I) - Z_I \neq \emptyset$. In order to use some symmetry arguments, let $\mathbf{A}^1 = X_I, \mathbf{A}^2 = Y_I$ and $\mathbf{A}^3 = Z_I$. Then, by $z \in (X_I \cup Z_I) - Y_I, v \in (Y_I \cup Z_I) - X_I$, and the assumption that $(X_I \cup Y_I) - Z_I \neq \emptyset$, we obtain $(A^i \cup A^j) - A^\ell \neq \emptyset$ whenever $\{i, j, \ell\} = \{1, 2, 3\}$. Since \mathbf{X}, \mathbf{Y} and \mathbf{Z} are horrifying bisets whose wall is $\{p\}$, we have $\mathbf{d}'(A^i) \leq (2k + 1) - k = k + 1$ for $i \in \{1, 2, 3\}$.

Claim 5.2.2. $A^\ell \subseteq A^i \cup A^j$ whenever $\{i, j, \ell\} = \{1, 2, 3\}$.

Proof. Suppose that $A^\ell - (A^i \cup A^j) \neq \emptyset$ for some $\{i, j, \ell\} = \{1, 2, 3\}$. By $z \in X_I \cap Z_I \cap N_H(s), v \in Y_I \cap Z_I \cap N_H(s)$ and $u \in X_I \cap Y_I \cap N_H(s)$, we have $A^i \cap A^j \neq \emptyset$ and $\overline{\mathbf{d}'}(A^i \cup A^j, A^\ell) \geq |(A^i \cup A^j) \cap A^\ell \cap N_H(s)| \geq 2$. As $A^i \cap A^j \neq \emptyset$ and $(A^i \cup A^j) - A^\ell \neq \emptyset$, it follows that $\mathbf{d}'(A^i \cap A^j), \mathbf{d}'((A^i \cup A^j) - A^\ell) \geq k$. Then, as $\mathbf{d}'(A^i), \mathbf{d}'(A^j), \mathbf{d}'(A^\ell) \leq k + 1$, Proposition 2.2.4(a) and (b) yield

$$\begin{aligned} 3(k + 1) - k &\geq \mathbf{d}'(A^i) + \mathbf{d}'(A^j) - \mathbf{d}'(A^i \cap A^j) + \mathbf{d}'(A^\ell) \\ &\geq \mathbf{d}'(A^i \cup A^j) + \mathbf{d}'(A^\ell) \\ &= \mathbf{d}'((A^i \cup A^j) - A^\ell) + \mathbf{d}'(A^\ell - (A^i \cup A^j)) + 2\overline{\mathbf{d}'}(A^i \cup A^j, A^\ell) \\ &\geq 2k + 4, \end{aligned}$$

a contradiction. □

Claim 5.2.3. *For any $i \neq j \in \{1, 2, 3\}$, we have $\overline{\mathbf{d}'}(A^i, A^j) \leq 1$.*

Proof. Let ℓ be the remaining element in $\{1, 2, 3\} - \{i, j\}$. By assumption, $(A^i \cup A^\ell) - A^j \neq \emptyset$. By Claim 5.2.2, $A^\ell - (A^i \cup A^j) = \emptyset$. This yields $A^i - A^j = ((A^i \cup A^\ell) - A^j) - (A^\ell - (A^i \cup A^j)) \neq \emptyset$. Similarly, $A^j - A^i \neq \emptyset$. It follows that $d'(A^i - A^j), d'(A^j - A^i) \geq k$. As $d'(A_i), d'(A_j) \leq k + 1$, Proposition 2.2.4(b) yields $2\overline{d'}(A^i, A^j) = d'(A^i) + d'(A^j) - d'(A^i - A^j) - d'(A^j - A^i) \leq 2(k + 1) - 2k = 2$. \square

Claim 5.2.4. $d'(X_I \cup Y_I) \geq k + 2$.

Proof. As (H, c) has a complete admissible splitting off, there exist $x, y \in N_H(s)$ such that $(H'', c'') := ((H, c)_{z,x}^1)_{v,y}^1$ is $(2, k)$ -connected in V . Since $z \in X_I \cap Z_I, v \in Y_I \cap Z_I$ and $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ are horrifying, we obtain $x \notin X_I \cup Z_I$ and $y \notin Y_I \cup Z_I$. Then, by Claim 5.2.2, $x, y \notin X_I \cup Y_I$. If $x = p = y$, then $f_{(H'', c'')}(Z) = f(Z) - 2$, so, since (H'', c'') is $(2, k)$ -connected in V and \mathbf{Z} is horrifying, we have $2k \leq f_{(H'', c'')}(Z) = f(Z) - 2 \leq (2k + 1) - 2 = 2k - 1$, a contradiction. So one of x and y belongs to $V - (\mathbf{X} \cup \mathbf{Y})_O$ and hence $\mathbf{X} \cup \mathbf{Y}$ is a nontrivial biset with respect to V . Then, since $\mathbf{X} \cup \mathbf{Y}$ is not horrifying, we have $d'(X_I \cup Y_I) = f(\mathbf{X} \cup \mathbf{Y}) - |w(\mathbf{X} \cup \mathbf{Y})|k \geq (2k + 2) - k = k + 2$. \square

By Claim 5.2.2, we have $X_I \cup Y_I = X_I \cup Y_I \cup Z_I$ and every edge that contributes to $d'(X_I \cup Y_I \cup Z_I)$ also contributes to $\overline{d'}(A^i, A^j)$ for some $i \neq j \in \{1, 2, 3\}$. By $k \geq 2$, and Claims 5.2.4 and 5.2.3, we obtain $4 \leq k + 2 \leq d'(X_I \cup Y_I) = d'(X_I \cup Y_I \cup Z_I) \leq \sum_{i \neq j} \overline{d'}(A^i, A^j) \leq 3$, a contradiction. This finishes the case.

Case 5.2.2. $w(\mathbf{Z}) \neq \{p\}$.

By symmetry, we may suppose that $v \in Z_I$. As $z \in X_I, v \in N_H(s) - X_I, \mathbf{Z}$ blocks (sz, sv) and $w(\mathbf{Z}) \neq \{p\}$, we may apply Lemma 5.2.3 to obtain that $\mathbf{X} \cup \mathbf{Z}$ is horrifying. Since $z, v \in (\mathbf{X} \cup \mathbf{Z})_I$, $\mathbf{X} \cup \mathbf{Z}$ blocks (sz, sv) . As $v \in Y_I, z \in N_H(s) - Y_I$, we may apply Lemma 5.2.3 once more and obtain that either $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$ is horrifying or $w(\mathbf{X} \cup \mathbf{Z}) = w(\mathbf{Y})$. In the first case we are done, so we may suppose that $w(\mathbf{X} \cup \mathbf{Z}) = w(\mathbf{Y}) = \{p\}$. If $Y_I - (X_I \cup Z_I) = \emptyset$, we obtain that $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} = \mathbf{X} \cup \mathbf{Z}$ is horrifying.

We may therefore suppose that $Y_I - (X_I \cup Z_I) \neq \emptyset$. As $z \in (X_I \cup Z_I) - Y_I$, we obtain that $d'(Y_I - (X_I \cup Z_I)) \geq k$ and $d'((X_I \cup Z_I) - Y_I) \geq k$. Also, as $\mathbf{X} \cup \mathbf{Z}$ and \mathbf{Y} are horrifying bisets whose wall is $\{p\}$, we obtain $d'(X_I \cup Z_I) \leq k + 1$ and $d'(Y_I) \leq k + 1$. As $u, v \in (X_I \cup Z_I) \cap Y_I \cap N_H(s)$, Proposition 2.2.4 (b) yields $2(k + 1) \geq d'(X_I \cup Z_I) + d'(Y_I) = d'((X_I \cup Z_I) - Y_I) + d'(Y_I - (X_I \cup Z_I)) + 2\overline{d'}(X_I \cup Z_I, Y_I) \geq 2k + 4$, a contradiction. \square

5.2.3.2 Description of the algorithm

We are now ready to describe the algorithm in the form of a pseudocode. It first is initialized with the input capacitated graph and an empty biset \mathbf{X} . The main part of the algorithm consists of a while-loop in which maximal splitting offs are executed and \mathbf{X} is modified. In order to apply the structure found in Lemma 5.2.10, we need \mathbf{X} to be horrifying. Therefore, the first part of the while-loop in lines 3 to 7 is concerned with reinitializing \mathbf{X} with a horrifying biset if \mathbf{X} is not horrifying before the iteration. The main part from line 9 to 24 deals with the case when \mathbf{X} is horrifying. Algorithm 3 then performs up to two maximal splitting offs of pairs of edges incident to s whose choice depends on \mathbf{X} . If none of these two maximal splitting offs leads to the deletion of an edge incident to s , Algorithm 3 augments \mathbf{X} in a beneficial way. After the last iteration of the while-loop, Algorithm 3 outputs the obtained capacitated graph after deleting s .

Algorithm 3 Complete admissible splitting off

Input: A minimal even extension for $(2, k)$ -connectivity $(H = (V \cup s, E), c)$ of a capacitated graph (G, c_0) .

Output: A minimum $(2, k)$ -connected augmentation of (G, c_0) .

```
1  $X := (\emptyset, \emptyset)$ ;  
2 while  $|N_H(s)| \geq 2$  do  
3   if  $X$  is not horrifying then  
4     let  $u \neq v \in N_H(s)$ ;  
5      $(H, c) = (H, c)_{u,v}^{max}$ ;  
6     if  $c(su), c(sv) > 0$  then  
7       let  $X$  be a biset blocking  $(su, sv)$ ;  
8   else  
9     let  $u \in X_I \cap N_H(s)$ ;  
10    let  $v \in N_H(s) - X_O$ ;  
11     $(H, c) = (H, c)_{u,v}^{max}$ ;  
12    if  $c(su), c(sv) > 0$  then  
13      let  $Y$  be a biset blocking  $(su, sv)$ ;  
14      if  $X \cup Y$  is horrifying then  
15         $X = X \cup Y$ ;  
16      else  
17        if  $X_I \cap N_H(s) \subseteq Y_I$  then  
18           $X = Y$ ;  
19        else  
20          let  $z \in (X_I - Y_I) \cap N_H(s)$ ;  
21           $(H, c) = (H, c)_{v,z}^{max}$ ;  
22          if  $c(sv), c(sz) > 0$  then  
23            let  $Z$  be a biset blocking  $(sv, sz)$ ;  
24             $X = X \cup Y \cup Z$ ;  
25 return  $(H, c) - s$ ;
```

5.2.3.3 Analysis of the algorithm

This last section is dedicated to the analysis of Algorithm 3. We first give a collection of properties of the capacitated graphs and bisets obtained at intermediate steps of Algorithm 3. We then conclude the correctness and the running time of Algorithm 3.

Proposition 5.2.3. *The following hold for every iteration i of the while-loop starting in line 2:*

- (a) *All steps in iteration i are well-defined.*
- (b) *(H, c) is $(2, k)$ -connected in V and has a complete admissible splitting off after iteration i .*
- (c) *$f(X) \leq 2k + 1, |w(X)| \leq 1$ and $X_O \neq V$ after iteration i .*

Proof. By Lemma 5.2.9, (b) holds before iteration 1 and trivially (c) also holds. Inductively, we may suppose that (a), (b) and (c) hold for all iterations $1, \dots, i - 1$. We show that they also hold for iteration i :

(a): The choice of u and v in line 4 is justified by the fact that the while-condition in line 2 was satisfied. If X is horrifying, the choice of u in line 9 is justified and Lemma 5.2.4 justifies the choice of v in line 10. The choice of z in line 20 is justified by the fact

that the if-condition in line 17 was not satisfied. The horrifying bisets in lines 7,13 and 23 exist by Lemma 5.2.2.

(b): It follows immediately from the construction that $c(\delta_H(s))$ always remains even and (H, c) always remains $(2, k)$ -connected in V . By Lemma 5.2.9, no obstacle in (H, c) can ever be created. Now Theorem 5.2.2 yields that (H, c) has a complete admissible splitting off after iteration i .

(c): As splitting offs do not increase f , it suffices to prove that X either remains unchanged or is horrifying after iteration i . First suppose that the if-condition in line 3 is satisfied. If the if-condition in line 6 is not satisfied, X remains unchanged. Otherwise, X is replaced by a horrifying biset in line 7.

Now suppose that the else-case starting in line 8 is executed. If the if-condition in line 12 is not satisfied, X remains unchanged, so suppose otherwise. If the if-condition in line 14 is satisfied, X is replaced by a horrifying biset in line 15, so suppose otherwise. If the if-condition in line 17 is satisfied, X is replaced by a horrifying biset in line 18. So suppose that the else-case starting in line 19 is executed. If the if-condition in line 22 is not satisfied, X remains unchanged. Otherwise, $u \in X_I \cap N_H(s), v \in N_H(s) - X_O, Y$ blocks $(su, sv), z \in (X_I - Y_I) \cap N_H(s), Z$ blocks (sv, sz) and $X \cup Y$ is not horrifying. Together with (b), Lemma 5.2.10 yields that $X \cup Y \cup Z$ is horrifying, so X is replaced by a horrifying biset in line 24. \square

We now obtain the correctness of our algorithm as a simple corollary:

Theorem 5.2.4. *If Algorithm 3 terminates, it outputs a minimum $(2, k)$ -connected augmentation of (G, c_0) .*

Proof. By Theorem 5.2.2, it is sufficient to prove that Algorithm 3 executes a complete admissible splitting off of the input capacitated graph. Let (H, c) be the current capacitated graph after the last iteration of the while-loop. By construction, $|N_H(s)| \leq 1$. If $N_H(s)$ contains a single vertex u , by Proposition 5.2.2 (d), (su, su) is not admissible in the input capacitated graph. As (H, c) has been obtained by admissible splitting offs, (su, su) is neither admissible in (H, c) . It follows that (H, c) does not have a complete admissible splitting, a contradiction to Proposition 5.2.3 (b). Hence s is an isolated vertex in (H, c) and so, by Proposition 5.2.3 (b), Algorithm 3 executed a complete admissible splitting off. \square

The remaining part is concerned with the running time analysis of Algorithm 3.

Theorem 5.2.5. *Algorithm 3 runs in $O(n^3(m + n \log n))$ time.*

Proof. Obviously the initialization of Algorithm 3 and the final output can be executed efficiently. Also, it follows from Lemma 5.2.8 and the fact that we can check in $O(m)$ time whether a given biset is horrifying that every iteration of the while-loop starting in line 2 can be executed in $O(n^2(m + n \log n))$ time. It remains to show that the while-loop runs a linear number of times. In order to do this, we define the parameter $M = |N_H(s)| + |N_H(s) - X_O|$. The decrease of M measures the progress of our algorithm. We next prove two claims that show that M regularly decreases.

Claim 5.2.5. *If in an iteration i the else-case starting in line 8 is executed, then M decreases in iteration i .*

Proof. If the if-condition in line 12 is not satisfied, $|N_H(s)|$ decreases in iteration i and X remains unchanged, so suppose otherwise. If the if-condition in line 14 is satisfied, then in line 15 X is replaced by a biset containing X and v leaves $N_H(s) - X_O$, so $|N_H(s) - X_O|$ decreases and $N_H(s)$ remains unchanged. Otherwise, by Lemma 5.2.3, we have $w(X) =$

$w(\mathbf{Y})$. Therefore, if the if-condition in line 17 is satisfied, \mathbf{X} is replaced by \mathbf{Y} in line 18 and \mathbf{Y} satisfies $X_O \cup \{v\} \subseteq Y_O$, so $|N_H(s) - X_O|$ decreases and $N_H(s)$ remains unchanged. So suppose that the else-case starting in line 19 is executed. If the if-condition in line 22 is not satisfied, $|N_H(s)|$ decreases in iteration i and \mathbf{X} remains unchanged, so suppose otherwise. Then in line 23 \mathbf{X} is replaced by a biset containing \mathbf{X} and v leaves $N_H(s) - X_O$, so $|N_H(s) - X_O|$ decreases and $N_H(s)$ remains unchanged. \square

Claim 5.2.6. *If in an iteration i the if-condition in line 3 is satisfied, then either M decreases in iteration i or M remains unchanged in iteration i and decreases in iteration $i + 1$.*

Proof. If the if-condition in line 6 is not satisfied, $|N_H(s)|$ decreases in iteration i and \mathbf{X} remains unchanged, so suppose otherwise. By Proposition 5.2.3(c), $|X_O \cap N_H(s)| \leq 2$ before iteration i . As \mathbf{X} is replaced by a horrifying biset in line 7, we have $|X_O \cap N_H(s)| \geq 2$ after iteration i . As $N_H(s)$ remains unchanged, M does not increase in iteration i . As \mathbf{X} is horrifying after iteration i , it follows that in iteration $i + 1$ the else-case starting in line 8 is executed, so M decreases in iteration $i + 1$ by Claim 5.2.5. \square

Claims 5.2.5 and 5.2.6 show that M never increases and decreases in at least one of two consecutive iterations. Further observe that M is always an integer satisfying $0 \leq M \leq 2n$. It follows that the while-loop runs at most $4n$ times. This finishes the proof. \square

Chapter 6

Conclusion

In this thesis, we have contributed to 3 different aspects of the theory of graph connectivity.

In the first part, we have dealt with orientations for arc-connectivity and vertex-connectivity. For arc-connectivity, this problem setting is relatively well understood due to Theorems 3.1.2, 3.1.3 and 3.1.6. While Theorems 3.1.2 and 3.1.3 are well integrated into the theory of connectivity, Theorem 3.1.6 has a somewhat isolated position. Our negative results in Theorems 3.1.19 and 3.1.20 confirmed the difficulty of relating Theorem 3.1.6 to the remaining theory. On the other hand, for Theorems 3.1.2 and 3.1.3, further generalizations can be considered. For the main problem considered in Section 3.1.3, the question whether this generalization exists remains open. However, we have given some approaches, both for the positive and the negative side. Another such generalization is the introduction of the Frank number in Section 3.1.4. Here, we have managed to develop some theory. Nevertheless, several important questions remain open.

For vertex-connectivity, much less is known. Despite the results of Thomassen and Durand de Gevigney, several fundamental problems are still unsolved, in particular Conjecture 3.2.2. With the consideration of good graphs, we deal with a new class of problems. Even though the classes we could identify are very limited, the depth of our proofs is significant. The identification of further classes of good graphs is another possible direction of research.

In the second part, we have dealt with arborescence packings. Since Edmonds proved Theorem 4.1.1, numerous generalizations of Theorem 4.1.1 have been developed. One of them is the consideration of reachability arborescences instead of spanning arborescences. We provided a new inductive method to deal with reachability arborescence packings. This allowed for a new proof of Theorem 4.2.1 and to settle the last open case in a series of generalizations involving mixed hypergraphs, reachability arborescences and matroid-based packings. Further, we showed that the connection between matroid intersection and arborescence packing that was observed by Edmonds can also be exploited in a much more general setting. This allowed to prove Theorem 4.3.3 that deals with packings of mixed hyperarborescences with flexible roots. Further, we considered a different way of generalizing Theorem 4.1.1 in which an additional balance condition was imposed on the spanning arborescences. While this problem is not tractable in general, an algorithm certifying that the problem is FPT for the case when at least a minimum of balance is required could be provided.

Finally, we dealt with the problem of $(2, k)$ -connectivity augmentation. While the properties of $(2, k)$ -connected graphs were deeply studied by Durand de Gevigney and Szigeti, a fast algorithm for $(2, k)$ -connectivity augmentation was missing in their work. Using the structure provided by them, we managed to fill this gap.

Bibliography

- [1] R. Aharoni, C. Thomassen, *Infinite, highly connected digraphs with no two arc-disjoint spanning trees*, Journal of Graph Theory, 13(1): 71-74, 1989,
- [2] P. Bachmann, *Analytische Zahlentheorie*, Teubner, Leipzig, 1894,
- [3] J. Bang-Jensen, S. Bessy, *(Arc-)disjoint flows in networks*, Theoretical Computer Science, 526: 28-40, 2014,
- [4] J. Bang-Jensen, A. Frank, B. Jackson, *Preserving and increasing local edge-connectivity in mixed graph*, SIAM Journal on Discrete Mathematics, 8(2): 155-178, 1995,
- [5] J. Bang-Jensen, G. Gutin, *Digraphs: Theory, Algorithms and Applications*, 2nd edition, Springer-Verlag, 2009,
- [6] J. Bang-Jensen, F. Havet, A. Yeo, *The complexity of finding arc-disjoint branching flows*, Discrete Applied Mathematics, 209: 16-26, 2016,
- [7] J. Bang-Jensen, J. Huang, X. Zhu, *Completing orientations of partially oriented graphs*, Journal of Graph Theory, 87(3): 285-304, 2018,
- [8] J. Bang-Jensen, A. Yeo, *Balanced branchings in digraphs*, Theoretical Computer Science, 595: 107-119, 2015,
- [9] K. Bérczi, A. Frank, *Packing arborescences*, Lecture Notes, RIMS Kokyuroku Bessatsu B23: Combinatorial Optimization and Discrete Algorithms, ed. S. Iwata, 1-31, 2010,
- [10] K. Bérczi, A. Frank, *Variations for Lovász' submodular ideas*, in: M. Grötschel, G.O.H. Katona (eds.), Building bridges between mathematics and Computer Science, Bolyai Society, Series: Mathematical Studies, 19, Springer, 137-164, 2008,
- [11] A. R. Berg and T. Jordán, *Two-connected orientations of Eulerian graphs*, Journal of Graph Theory, 52(3): 230-242, 2006,
- [12] P. Berman, M. Karpinski, A. Scott, *Approximation hardness of short symmetric instances of MAX-3-SAT*, Electronic Colloquium on Computational Complexity, Technical Report TR03-049, 2003,
- [13] A. Bernáth, S. Iwata, T. Király, Z. Király, Z. Szigeti, *Recent results on well-balanced orientations*, Discrete Optimization, 5:663-676, 2008,
- [14] A. Bernáth, G. Joret, *Well-balanced orientations of mixed graphs*, Information Processing Letters, 106(4): 149-151, 2008,
- [15] S. Bessy, personal communication, 2020,

- [16] S. Bessy, F. Hörsch, A. K. Maia, D. Rautenbach, I. Sau, *FPT algorithms for packing k -safe spanning rooted sub(di)graphs*, <https://arxiv.org/abs/2105.01582>,
- [17] F. Boesch, R. Tindell, *Robbin's theorem for mixed multigraphs*, American Math. Monthly, 87: 716-719, 1980,
- [18] G. R. Cai, Y. G. Sun, *The minimum augmentation of any graph to k -edge-connected graphs*, Networks, 19:151-172, 1989,
- [19] J. Cheriyan, personal communication, 2017,
- [20] J. Cheriyan, O. Durand de Gevigney, Z. Szigeti, *Packing of rigid spanning subgraphs and spanning trees*, Journal of Combinatorial Theory, Series B, 105 : 17–25, 2014,
- [21] S. Cook, *The complexity of theorem proving procedures*, Proceedings of the Third Annual ACM Symposium on Theory of Computing, 151–158, 1971,
- [22] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, *Parameterized Algorithms*, Springer, 2015,
- [23] M. DeVos, T. Johnson, P. Seymour, *Cut Coloring and Circuit Covering*, manuscript, <https://web.math.princeton.edu/~pds/papers/cutcolouring/paper.pdf>,
- [24] O. Durand de Gevigney, *On Frank's conjecture on k -connected orientations*, Journal of Combinatorial Theory, Series B, 141: 105-114, 2020,
- [25] O. Durand de Gevigney, personal communication, 2017,
- [26] O. Durand de Gevigney, V. H. Nguyen, Z. Szigeti, *Matroid-based packing of arborescences*, SIAM Journal on Discrete Mathematics, 27(1), 567-574, 2013,
- [27] O. Durand de Gevigney, Z. Szigeti, *On $(2, k)$ -connected graphs*, Journal of Graph Theory, 91(4): 305-325, 2019,
- [28] J. Edmonds, *Edge-disjoint branchings*, in: B. Rustin (ed.), Combinatorial Algorithms, Academic Press, New York, 91-96, 1973,
- [29] J. Edmonds, *Matroid intersection*, Annals of Discrete Mathematics, 4: 39-49, 1979,
- [30] J. Edmonds, *Submodular functions, matroids, and certain polyhedra*, in: eds. R. Guy, H. Hanani, N. Sauer, and J. Schönheim, Combinatorial Structures and their Applications, Gordon and Breach, New York, 69-70, 1970,
- [31] J. Edmonds, *Some well-solved problems in combinatorial optimization*, in: Combinatorial Programming: Methods and Applications (Proceedings NATO Advanced Study Institute, Versailles, 1974. B. Roy, ed.), Reidel, Dordrecht, 285-301, 1975,
- [32] J. Edmonds, D. R. Fulkerson, *Transversals and matroid partition*, J. Res. Nat. Bur. Standards, B69, 147-153, 1965,
- [33] Egres Open Problems, Problem:"Smooth well-balanced orientations with prescribed in-degrees",http://lemon.cs.elte.hu/egres/open/Smooth_well-balanced_orientations_with_prescribed_in-degrees,
- [34] Egres Open Problems, Problem:"Undirected node-connectivity augmentation",http://lemon.cs.elte.hu/egres/open/Undirected_node-connectivity_augmentation,

- [35] Egres Open Problems, Problem:” *Well-balanced orientations of hypergraphs*”, http://lemon.cs.elte.hu/egres/open/Well-balanced_orientations_of_hypergraphs,
- [36] K. P. Eswaran, R. E. Tarjan, *Augmentation problems*, SIAM Journal on Computing, 5(4): 653-665,1976,
- [37] L. R. Ford, D. R. Fulkerson, *Flows in Networks*, Princeton University Press, PrincetonNJ., 1962,
- [38] Q. Fortier, Cs. Király, M. Léonard, Z. Szigeti, A. Talon, *Old and new results on packing arborescences in directed hypergraphs*, Discrete Applied Mathematics, 242: 26-33, 2018,
- [39] A. Frank, *Applications of submodular functions*, in: ed. K. Walker, Surveys in Combinatorics, London Mathematical Society Leteure Note Series 187, Cambridge University Press, Cambridge, 85-136, 1993,
- [40] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM Journal on Discrete Mathematics, 5(1): 22-53, 1992,
- [41] A. Frank, *Connections in Combinatorial Optimization*, Oxford University Press, 2011,
- [42] A. Frank, *Connectivity and network flows*, in: Handbook of Combinatorics, Elsevier, Amsterdam, 117-177, 1995,
- [43] A. Frank, *Kombinatorikus algoritmusok, algoritmikus bizonyítások*, Egyetemi doktori értekezés, ELTE TTK, Budapest (PhD thesis),
- [44] A. Frank, *On disjoint trees and arborescences*, in: Algebraic Methods in Graph Theory, Colloquia Mathematica Soc. J. Bolyai, 25:159-169, 1978,
- [45] A. Frank, Personal communication, 2018.
- [46] A. Frank, T. Jordán, *Directed vertex-connectivity augmentation*, Mathematical Programming, 84: 537-553, 1999,
- [47] A. Frank, T. Jordán, *Minimal Edge-Coverings of Pairs of Sets*, Journal of Combinatorial Theory, Series B, 65(1): 73-110, 1995,
- [48] A. Frank, T. Jordán, Z. Szigeti, *An orientation theorem with parity conditions*, Discrete Applied Mathematics 115: 37-45, 2001,
- [49] A. Frank and Z. Király, *Graph orientations with edge-connection and parity constraints*, Combinatorica 22: 47-70, 2002,
- [50] A. Frank, T. Király, Z. Király, *On the orientation of graphs and hypergraphs*, in: ed. S. Fujishige, Discrete Applied Mathematics, 131(2): 385-400, 2003,
- [51] A. Frank, T. Király, M. Kriesell, *On decomposing a hypergraph into k connected sub-hypergraphs*, Discrete Applied Mathematics, 131: 373-383, 2003,
- [52] H. N. Gabow, *Efficient splitting off algorithms for graphs*, in: STOC '94: Proceedings of the Twenty-Sixth AnnualACM Symposium on Theory of Computing, ACM Press: 696-705, 1994,

- [53] A. Gács, T. Héger, Z. Weiner, *On regular graphs of girth six arising from projective planes*, 34(2): 285-296, 2013,
- [54] H. Gao, D. Yang, *Packing of maximal independent mixed arborescences*, Discrete Applied Mathematics, 289: 313-319, 2021,
- [55] H. Gao, D. Yang, *Packing of spanning mixed arborescences*, <https://arxiv.org/abs/2005.03218>,
- [56] M. Garey, D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, 1979,
- [57] A. M. H. Gerards, *On 2-vertex connected orientations*, unpublished manuscript,
- [58] S. L. Hakimi, *On the degrees of the vertices of a directed graph*, J. Franklin Inst 279(4): 290-308, 1965,
- [59] F. Hörsch, *Checking the admissibility of odd-vertex pairings is hard*, <https://arxiv.org/abs/2103.00306>,
- [60] F. Hörsch, Z. Szigeti, *Connectivity of orientations of 3-edge-connected graphs*, European Journal of Combinatorics, 94, 103292, 2021,
- [61] F. Hörsch, Z. Szigeti, *Eulerian orientations and vertex-connectivity*, Discrete Applied Mathematics, 289: 115-124, 2021,
- [62] F. Hörsch, Z. Szigeti, *Packing of mixed hyperarborescences with flexible roots via matroid intersection*, <https://arxiv.org/abs/2012.13899>, accepted to Electronic Journal of Combinatorics,
- [63] F. Hörsch, Z.Szigeti, *Reachability in arborescence packings*, <https://arxiv.org/abs/2006.16190>,
- [64] F. Hörsch, Z. Szigeti, *The $(2, k)$ -connectivity augmentation problem: Algorithmic aspects*, <https://arxiv.org/abs/2004.08645>, to appear in Algorithmica,
- [65] A. Huck, *Disproof of a conjecture about independent branchings in k -connected directed graphs*, Journal of Graph Theory, 20: 235-239, 1995,
- [66] A. Huck, *Independent branchings in acyclic digraphs*, Discrete Mathematics, 199: 245-249, 1999,
- [67] R. Impagliazzo, R. Paturi, *On the complexity of k -SAT*, Journal of Computer and System Sciences, 62(2): 367–375, 2001,
- [68] R. Impagliazzo, R. Paturi, F. Zane, *Which problems have strongly exponential complexity?*, Journal of Computer and System Sciences, 63(4): 512–530, 2001,
- [69] B. Jackson, T. Jordán, *Independence free graphs and vertex connectivity augmentation*, Journal of Combinatorial Theory, Series B, 94(1): 31-77, 2005,
- [70] A. Joó, *Edmonds branching theorem in digraphs without forward-infinite paths*, Journal of Graph Theory, 83(3): 303-311, 2016,
- [71] N. Kamiyama, N. Katoh, A. Takizawa, *Arc-disjoint in-tress in digraphs*, Combinatorica, 29(2): 197-214, 2009,

- [72] A. Kaneko, K. Ota. *On minimally (n, λ) -connected graphs*, Journal of Combinatorial Theory, Series B,, 80(1): 156-171, 2000,
- [73] R. M. Karp, *Reducibility Among Combinatorial Problems*, In R. E. Miller; J. W. Thatcher; J.D. Bohlinger (eds.). Complexity of Computer Computations. New York: Plenum., 85–103, 1972,
- [74] N. Katoh, S. Tanigawa, *Rooted-tree decompositions with matroid constraints and the infinitesimal rigidity of frameworks with boundaries*, SIAM Journal on Discrete Mathematics, 27: 155-185, 2013,
- [75] A. B. Kempe, *A memoir on the theory of mathematical form*, Philos. Trans. R. Soc., 177, 1-70, 1886,
- [76] Cs. Király, *On maximal independent arborescence packing*, SIAM Journal on Discrete Mathematics, 30(4): 2107-2114, 2016,
- [77] Cs. Király, personal communication,
- [78] Cs. Király, unpublished manuscript,
- [79] Cs. Király, Z. Szigeti, S. Tanigawa, *Packing of arborescences with matroid constraints via matroid intersection*, Mathematical Programming, 181: 85-117, 2020,
- [80] T.Király, J.Szabó, *A note on parity constrained orientations*, Combinatorica, 29(5):619-628, 2009,
- [81] Z. Király, Z. Szigeti, *Simultaneous well-balanced orientations of graphs*, Journal of Combinatorial Theory, Series B, 96(5):684-692, 2006,
- [82] D. König, *Theory of finite and infinite graphs*, Birkhäuser, 1990,
- [83] D. König, *Gráfok és alkalmazásuk a determinánsok és a halmazok elméletére*, Matematikai és Természettudományi Értesítő, 34: 104-119, 1916,
- [84] B. Korte, J. Vygen, *Combinatorial Optimization, Theory and Algorithms*, Springer, Fifth edition, 2012,
- [85] M. Levit, L. S. Chandran, J. Cheriyan, *On eulerian orientations of even-degree hypercubes*, Operations research Letters, 46(5): 553–556, 2018,
- [86] M. Lorea, *Hypergraphes et matroides*, Cahiers Centre Etudes Rech. Oper. 17: 289-291, 1975,
- [87] L. Lovász, *Combinatorial Problems and Exercices*, North-Holland, Amsterdam, 1979,
- [88] L. Lovász, *Connectivity in digraphs*, Journal of Combinatorial Theory, Series B, 15 : 174-177, 1973,
- [89] W. Mader, *A reduction method for edge-connectivity in graphs*, Annals of Discrete Mathematics, 3:145-164, 1978,
- [90] W. Mader, *Konstruktion aller n -fach kantenzusammenhängenden Digraphen*, European Journal of Combinatorics, 3 : 63-67, 1982,
- [91] J. C. Martin, *Introduction to Languages and the Theory of Computation*, McGraw-Hill, 1990,

- [92] T. Matsuoka, S. Tanigawa, *On reachability mixed arborescence packing*, Discrete Optimization, 32: 1-10, 2019,
- [93] K. Menger, *Zur allgemeinen Kurventheorie*, Fund. Math. 10, 96-115, 1927,
- [94] H. Nagamochi, T. Ibaraki, *Graph connectivity and its augmentation: applications of MA ordering*, Discrete Applied Mathematics, 123: 447-472, 2002,
- [95] C. St. J. A. Nash-Williams, *Edge-disjoint spanning trees of finite graphs*, Journal of the London Mathematical Society, 36:445–450, 1961,
- [96] C.St.J.A. Nash–Williams, *On orientations, connectivity, and odd vertex pairings in finite graphs*, Canad. J. Math., 12: 555–567, 1960,
- [97] Open Problem Garden, Problem:” *The Berge-Fulkerson Conjecture*”,http://garden.irmacs.sfu.ca/op/the_berge_fulkerson_conjecture,
- [98] J. Oxley, *Matroid Theory*, Second edition, Oxford University Press, New York, 2011,
- [99] J. Petersen, *Die Theorie der regulären graphs*, Acta Mathematica, 15: 193–220, 1891,
- [100] H. E. Robbins, *A theorem on graphs with an application to a problem of traffic control*, American Math. Monthly 46, 281-283, 1939,
- [101] T. J. Schaefer, *The Complexity of Satisfiability Problems*, in: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing, STOC '78, 3:216-226, 1978,
- [102] P. D. Seymour, *On multi-colourings of cubic graphs, and conjectures of Fulkerson and Tutte*, Proc. London Math. Soc., 38(3):423-460, 1979,
- [103] Z. Szigeti, *A survey on arborescence packings*, unpublished manuscript,
- [104] C. Thomassen, *Configurations in graphs of large minimum degree, connectivity, or chromatic number*, Ann. New York Acad. Sci., 555, 402-412, 1989,
- [105] C. Thomassen, *Strongly 2-connected orientations of graphs*, Journal of Combinatorial Theory, Series B, 110:67-78, 2015,
- [106] W. T. Tutte, *On the problem of decomposing a graph into n connected factors*, Journal of the London Mathematical Society, 36(1):221-230, 1961,
- [107] L. Végh, A. Benczúr, *Primal-dual approach for directed vertex connectivity augmentation and generalizations*, ACM Transactions on Algorithms, 20, 2008,
- [108] K. Vidyasankar, *Covering the edge set of a directed graph with trees*, Discrete Mathematics, 24(1): 79-85, 1978,
- [109] V. G. Vizing, *On an estimate of the chromatic class of a p -graph*, Metody Diskret. Analiz., 3:25-30, 1964,
- [110] T. Watanabe, A. Nakamura, *Edge-connectivity augmentation problems*, Comp. System Sci., 35:96-144, 1987,
- [111] H. Whitney, *On the abstract properties of linear independence*, Amer. J. Math., 57: 509-533, 1935,
- [112] R. W. Whitty, *Vertex-disjoint paths and edge-disjoint branchings in directed graphs*, Journal of Graph Theory, 11: 349-358, 1987.