Univerzita Karlova v Praze
Matematicko–fyzikální fakulta

DIPLOMOVÁ PRÁCE

Marek Sulovský

Pakování T–cest
Packing T–paths

Katedra aplikované matematiky

Vedoucí diplomové práce	Prof. RNDr. Jiří Matoušek, DrSc.
Odborný konzultant	Prof. Alexander Schrijver

Studijní program	Informatika
Studijní obor	Diskrétní matematika a optimalizace

Acknowledgments

This work is mainly based on my master thesis "Packing T–paths" submitted at Vrije Universiteit in Amsterdam under supervision of Prof. Alexander Schrijver. I would like to express my thanks to the people who supported me while working on this thesis, either in Amsterdam, or later while improving it.

First of all I would like to thank my adviser, Lex Schrijver, for the amazing opportunity to work on such an interesting topic with his support and for sharing his knowledge of this topic with me. It has been a great pleasure and experience for me to work with him.

I am also grateful to my supervisor, Jiří Matoušek, for supporting my decision to go to study Amsterdam. I would also like to thank him for recommending Lex Schrijver's group to me and helping me have the opportunity to start working on my thesis there. But most of all, I would like to thank him for for being relly helpful all the time.

I would like to thank Evert Wattel for being the second reader of the original work in Amsterdam and pointing out some mistakes.

I am also grateful to Jan Hladký and Martin Klimeš for various suggestions and comments.

Last but not least I would like to thank my family for their support in all times.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním své práce a jejím zveřejňováním.

V Zürichu dne 8. prosince 2006

Marek Sulovský

Contents

Název práce: Pakování T–cest
Autor: Marek Sulovský
Pracoviště: Katedra aplikované matematiky, Univerzita Karlova v Praze
Vedoucí práce: Prof. RNDr. Jiří Matoušek, DrSc.
Odborný konzultant: Prof. Alexander Schrijver, CWI Amsterdam
Abstrakt:
Disjunktní T–cesty, jedna z oblastí teorie grafů a kombinatorické optimalizace, určitým způsobem zobecňují párování a disjunktní cesty v grafech. Pro neorientovaný graf $G = (V, E)$ a množinu $T \subseteq V$, nazveme T–cestou každou cestu spojující dva různé vrcholy v T. V případě T–cest se zejména zajímáme o maximální počet disjunktních T–cest.

V této práci shrneme hlavní teoretické i algoritmické výsledky z této problematiky, zejména min–maxové věty (Gallaiovu a Maderovy věty o disjunktních T–cestách), algoritmy (algoritmy na maximální počet disjunktních cest pomocí matroidového párování a maximální počet hranově disjunktních T–cest pomocí elipsoidové metody).

Navíc prezentujeme vlastní formulaci problému maximálního počtu disjunktních T–cest jako lineární program a několik dalších dílčích pozorování o zobecněních T–cest.
Klíčová slova: pakování T-cest, Maderova věta o disjunktních cestách, lineární programování

Title: Packing T–paths
Author: Marek Sulovský
Department: Department of Applied Mathematics, Charles University in Prague
Supervisor: Prof. RNDr. Jiří Matoušek, DrSc.
Adviser: Prof. Alexander Schrijver, CWI Amsterdam
Abstract:
The disjoint T–paths, one of the areas in graph theory and combinatorial optimization, in a sense generalize matchings and disjoint paths in graphs. For an undirected graph $G = (V, E)$ and $T \subseteq V$, a T–path is a path connecting two different vertices of T. We are usually interested in the maximum number of disjoint T–paths.

In this thesis we compile the most important theoretical and algorithmic results from this area, especially min–max characterizations (Gallai's and Mader's disjoint T–paths theorems), algorithmic results (disjoint T-paths from matroid matching algorithm, edge–disjoint T–paths by the ellipsoid method).

Furthermore we derive a linear programming formulation of one of the maximum vertex–disjoint T–paths packing problem and presents few smaller observations concerning packings of T–paths generalizations.
Keywords: T-paths packing, Mader's disjoint paths theorem, linear programming

CHAPTER 1

Introduction

The goal of this thesis was to do some research in the topic of T–paths and their further generalizations, min–max relations concerning packing problems, their polyhedral descriptions or linear programming formulations. This text brings a linear programming formulation of one of those problems and some other minor new facts about T–paths.

The text was written in such a way that a person with some basic knowledge of mathematics and graph theory should understand it, therefore the basic necessary background facts are briefly given at the beginning.

In Chapter 2 we list the basic definitions and notations from various areas of mathematics needed. We recall the basic denotation, definitions and theorems from linear algebra, graph theory (graph basics, matchings, b–matchings, disjoint paths), matroid theory, complexity theory (classes P and NP) and linear optimization. Everything is explained briefly and is meant to make reader familiar with the notation used. The only exception is the section about b–matchings. They are not so commonly known and therefore we give more details.

The following part, Chapter 3, is a compilation of known theoretical results about T–paths from the book [**Sch03c**] and articles [**Sch01**], [**SS04**], [**CGG+04**], [**Pap05b**] and [**Pap06**]. It consists of a description of basic disjoint T–paths packing problems, of Gallai [**Gal61**] and Mader [**Mad78a**], [**Mad78b**] and relations between them. Then the min–max relations for those problems are stated. We also mention the relation of the path packings to matroids. This chapter also covers further generalizations of those problems, like e.g. non–zero T–paths in group–labeled graphs, and some known min–max results about them.

Chapter 4 is a compilation of algorithmic results relevant to T–paths from the book [**Sch03c**] and articles [**Lov80**], [**Lov81**], [**CCG04**] and [**Pap05a**]. It describes algorithmic view of disjoint T–paths packing problems and known algorithms for them. Two basic algorithms, the first one using graph matchings, which works only for one type of the problem, and the second one, Lovász's algorithm [**Lov80**], [**Lov81**] using matroid matchings in linear spaces are shown. Furthermore, recent combinatorial algorithmic results by Chudnovsky et. al. [**CCG04**] and Pap [**Pap05a**] are mentioned.

The last Chapter 5 consists most of all of the main outcomes of this work — a simple linear programming description of fully vertex–disjoint non–returning T–paths derived in different ways and some basic insight to properties of possible further generalizations of path packings like returning T–paths or non–returning $\mathcal{S}$–paths are shown.

CHAPTER 2

Preliminaries, definitions and denotation

1. Linear algebra

Some basic knowledge of linear algebra, vector spaces and matrices is required for understanding the following text. From now on we will consider all vectors as *column vectors.* The *components* of a vector $\mathbf{x} = (x_1, \dots, x_n)^T$ are $x_1, \dots, x_n$.

Functions $x : S \to \mathbb{R}$ are identified with the vectors $\mathbf{x}$ such that $x_s = x(s)$ for each $s \in S$.

For any $R \subseteq S$ the *incidence vector* of R is the vector χ_R defined as

$$\chi_R(s) := \begin{cases} 1 & \text{if } s \in R \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

For any $s \in S$ we denote

$$\chi_s := \chi_{\{s\}} \tag{2}$$

Given a vector space $\mathbb{R}^S$, the all–zero vector is denoted by $\mathbf{0}_S$ or just $\mathbf{0}$ and similarly all–one vector is denoted by $\mathbf{1}_S$ or $\mathbf{1}$.

Whenever we write any matrix or vector multiplication we implicitly assume compatibility of the dimensions.

For a function $x : S \to \mathbb{R}$ and any $R \subseteq S$, we denote

$$x(R) := \sum_{s \in R} x(s) \tag{3}$$

Vectors $x_1, \dots, x_k$ are called *linearly independent* if there are no $\lambda_1, \dots, \lambda_k$ such that $\sum_{i=1}^{k} \lambda_i x_i = \mathbf{0}$ and not all of λ_i are equal to 0. *Linear hull* of a set X of vectors is the set of all their linear combinations and is denoted by $\operatorname{lin.hull}(X)$, more precisely

$$\operatorname{lin.hull}(X) := \{\sum_{y \in Y} y\lambda_y, Y \subseteq X \text{ finite }, \lambda \in \mathbb{R}^Y\} \tag{4}$$

A set X is a *linear space* if $X = \operatorname{lin.hull}(Y)$ for some Y. Dimension of a space L is the maximum number of linearly independent vectors in that space and is denoted by $\dim(L)$. If X, Y are subsets of a linear space L and $z \in L$, $\lambda \in \mathbb{R}$, then

$$z + X := \{z + x \mid x \in X\} \tag{5a}$$

$$X + Y := \{x + y \mid x \in X, y \in Y\} \tag{5b}$$

$$\lambda X := \{\lambda x \mid x \in X\} \tag{5c}$$

If X, Y are subspaces of L then

$$X/Y := \{x + Y \mid x \in X\} \tag{6}$$

is a *quotient space*, which is a linear space, with addition and scalar multiplication given by (5). The dimension of a quotient space is known to be $\dim(X/Y) = \dim(X) - \dim(X \cap Y)$. The idea behind this definition is that we consider all vectors in the space $x + Y$ as equivalent and we are performing all operations on these equivalence classes.

2. Sets

For U, V sets we will sometimes use additive notation for set union and set subtraction so

$$U - V := U \setminus V \tag{7}$$

$$U + V := U \cup V \tag{8}$$

If it is also clear from the context that v is not a set but an element we will also use this notation for set and element

$$U - v := U \setminus \{v\} \tag{9}$$

$$U + v := U \cup \{v\} \tag{10}$$

A family $\mathcal{F}$ of sets is called a *packing* if the sets in $\mathcal{F}$ are pairwise disjoint.

Given a set X, we will denote $\binom{X}{n} := \{Y \subseteq X \mid |Y| = n\}$ set of all subsets of X of cardinality n. For $n \in \mathbb{N}$ we will denote $[n] = \{1, \dots, n\}$.

3. Permutations

Permutation π on a set S is a bijective mapping $\pi : S \longrightarrow S$. We will usually talk about permutations on $[k]$. *Identity* on S is the permutation on S given by $\mathrm{id}_S : x \longrightarrow x$ sometimes also denoted just by id. A *transposition* is a permutation equal to identity for all but two elements of S. We denote transpositions by (a, b) so that

$$(a, b)(x) := \begin{cases} b & x = a \\ a & x = b \\ x & \text{otherwise} \end{cases} \tag{11}$$

4. Linear programming

Linear programming is a problem of optimizing a linear function over a linearly bounded region of a space. It is a quite powerful tool and has many application both in solving "real–life" problems as well as in many areas of combinatorics, for its theoretical properties. For a detailed reading on this topic we would recommend [**Sch86**].

Linear programming is a special case of optimization problem.

4.1. Optimization. Optimization problem is a problem of finding

$$\min\{f(x) \mid x \in S\} \tag{12}$$

given a set S and a function $f : S \longrightarrow \mathbb{R}$. We call elements of S *feasible solutions* of our problem. There are following possibilities:

- $S = \emptyset$: we call (12) *infeasible*,
- $\inf\{f(x) \mid x \in S\} = -\infty$: we call (12) *unbounded*,
- $\inf\{f(x) \mid x \in S\} > -\infty$ but the minimum does not exist,
- $\min\{f(x) \mid x \in S\}$ exists.

4.2. Polytopes and polyhedra. Let $x_1, \ldots, x_k \in \mathbb{R}^n$. Vector $y \in \mathbb{R}^n$ is called a *convex combination* of $x_1, \ldots, x_n$ if there are $\lambda_1, \ldots, \lambda_k \geq 0$ such that $y = \sum_{i=1}^{k} \lambda_i x_i$ and $\sum \lambda_i = 1$. For a set of vectors X we call its *convex hull* the set of all finite convex combinations of vectors from X and denote

$$\text{conv. hull}(X) = \{\sum_{y \in Y} y\lambda_y \mid \mathbf{1}^T\lambda = 1, \lambda \in \mathbb{R}_+^Y, Y \subseteq X \text{ finite } \} \tag{13}$$

A *convex set* is a set $S \subseteq \mathbb{R}^n$, such that for each $x, y \in S$ also $\lambda x + (1-\lambda)y \in S$ for all $\lambda \in [0,1]$. Convex hull of X is then the minimal convex set containing X.

A *polyhedron* P is a subspace of $\mathbb{R}^n$ given by

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\} \tag{14}$$

where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$. We call a set P polytope if $P = \text{conv. hull}(X)$ for some finite set of vectors X. We call polyhedron P bounded if there is a number $s \in \mathbb{R}_+$, such that $\forall x \in P$: $\|x\|_\infty \leq s$. The following theorem describes the relation between polytopes and polyhedra.

THEOREM 2.1. *Every polytope is a polyhedron and every bounded polyhedron is a polytope.*

It is straightforward to see from the definition of polyhedron that if P is a polyhedron and $x \notin P$ then x violates at least one of the defining inequalities $A_i x \leq b_i$.

4.3. Linear optimization. *Linear program* is an optimization problem of optimizing linear function c over a polyhedron P

$$\max\{c^T x \mid x \in P\} \tag{15}$$

There are many equivalent forms of formulating such a problem: minimization, maximization and the way the inequalities are formed. For more detail see the lecture notes [**Pen**] or introductory part of [**Sch03a**] (or any other basic text on linear programming). We will use a standard form:

$$\max\{c^T x \mid Ax \le b, x \ge 0, x \in \mathbb{R}^n\} \tag{16}$$

where $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^n$. The c above is called an *objective function.* Linear programming is a special case of optimization problems and it is always either infeasible, unbounded or the optimum exists. Reader can easily see following fact:

OBSERVATION 2.2. *If the maximum m of (16) exists and the polyhedron $P := \{x \mid Ax \le b\}$ form (16) is bounded, then m is attained by a vertex of P.*

This is useful when we tackle various combinatorial problems with linear programming methods as we will see later.

4.4. Duality. *Dual program* to a linear program (16) is the linear program

$$\min\{y^T b \mid y^T A \ge c^T, y \ge 0, y \in \mathbb{R}^m\} \tag{17}$$

There is a relation between (16) and (17) called *strong duality*:

THEOREM 2.3 (Strong duality). *For any matrix $A \in \mathbb{R}^{m\times n}$ and vectors $b \in \mathbb{R}^m, c \in \mathbb{R}^n$*

$$\max\{c^T x \mid Ax \le b, x \ge 0, x \in \mathbb{R}^n\} = \min\{y^T b \mid y^T A \ge c^T, y \ge 0, y \in \mathbb{R}^m\} \tag{18}$$

as long as both problems are feasible.

There is a characterization of primal and dual optimal solutions:

THEOREM 2.4 (Complementary slackness). *Let $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$. Let $a_1, \dots a_m$ be the rows of A. If x is a feasible solution of (16) and y is a feasible solution of (17) then the following conditions are equivalent*

(19a) *x and y are both optimal solutions*

(19b) *for each $i \in [m]$ either $y_i = 0$ or $a_i x = b_i$*

4.5. Ellipsoid method. A breakthrough in the theory of linear programming was discovery of a polynomial–time algorithm by [**Kha79**] and later consequences for many combinatorial problems described in [**GLS81**].

THEOREM 2.5 (Ellipsoid method). *A linear program $\max\{c^T x \mid Ax \le b, x \ge 0, x \in \mathbb{R}^n\}$ with $A \in \mathbb{Z}^{m\times n}, c \in \mathbb{Z}^n, b \in \mathbb{Z}^m$ can be solved in time polynomial to n, m and T where T is the maximal absolute value of entries in A, b.*

Restricting to $\mathbb{Z}$ is the same as restricting to $\mathbb{Q}$ and multiplying all the values by the least common multiple of their denominators. This method can be even extended to provide more interesting characteristics.

Let P be a polyhedron in $\mathbb{R}^n$. *Separation problem* for P is: Given a vector $y \in \mathbb{Q}^n$ decide whether y belongs to P or not, and in the latter case, find a vector $a \in \mathbb{Q}^n$ such that $ax < ay$ for all $x \in P$.

Ellipsoid method can be modified and is often used in a way that its number of elementary operations is polynomial in n and T while some of these operations can be solving a separation problem for the original polytope. Thus if we can solve the separation problem polynomially in the dimension of the polytope we do not have to care how many inequalities describe it. This is especially used for combinatorial problems where there are quite often exponentially many inequalities describing them but they are sometimes well structured and can be separated in polynomial time — as for example in Keijsper, Pendavingh, Stougie [**KPS06**].

5. Graphs

A *graph* is a pair $G = (V, E)$ where V is a finite set and E is a family of unordered pairs from V. Elements of V are called *vertices* and elements of E *edges*. We use the following shorthand notation for edges

$$uv = \{u, v\} \tag{20}$$

We denote $V(G)$ the set of vertices of G and $E(G)$ the family of edges of G. We have chosen the definition carefully so it allows multiple parallel edges and loops (edges of the form vv). A graph without multiple edges and loops is called *simple graph.* We call G *bipartite* if there are $X, Y \subseteq V$ such that $X \cup Y = V$ and $X \cap Y = \emptyset$ and all edges $e \in E$ have $e \cap X \neq \emptyset$ and $e \cap Y \neq \emptyset$. Examples of graphs are *complete graph on n vertices* $K_n := ([n], \binom{[n]}{2})$, *path of length n* $P_n := ([n+1], \{\{i, i+1\} \mid i \in [n]\})$, *circuit of length n* defined by $C_n := ([n], \{\{i, (i \bmod n) + 1\} \mid i \in [n]\})$ and *complete bipartite graph* $K_{m,n} := ([n] \cup [m]', \{\{i, j'\} \mid i \in [n], j \in [m]\})$. For a simple graph G we call $\overline{G} := (V(G), \binom{V(G)}{2} - E(G))$ a *complement* of G.

Vertices $u, v \in V$ are called *adjacent* if there is an edge $uv \in E$. Vertex u is called *incident* to an edge $e \in E$ if $u \in e$. Given a graph $G = (V, E)$ and a vertex $v \in V$ we call $N_G(v) = N(v) := \{u \in V \mid \{u, v\} \in E\}$ *set of neighbours* of v. A *subgraph* of G is an arbitrary graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. For a set $U \subseteq V$ we denote $E[U] := \{e \in E \mid e \subseteq U\}$. $G[U] := (U, E[U])$ is called a *subgraph induced by U*. For $U, W \subseteq V, U \cap W = \emptyset$ we denote

$$\delta(U, W) := \{e \in E \mid |U \cap e| = |W \cap e| = 1\} \tag{21}$$

By $\delta(U)$ we denote the family of edges leaving U and by $d(U)$ we denote $|\delta(U)|$.

A path in G is a sequence $P := (s = v_0 e_0 v_1 \dots e_{k-1} v_k = t)$ where all $v_i \in V$, all $e_i \in E$, $e_i = v_i v_{i+1}$ and no vertex appears more than once. We say that P *connects* s with t and call it an s–t path. For $u, v \in V$ we say that u is *reachable* from v

if there exists a u–v path. A *connected component* or *component of connectivity* in G is a non–empty set of vertices of G such that it contains some $v \in V$ and consists of all vertices that are reachable from v. A graph is *connected* if it has a single connected component. Otherwise we call it *disconnected*. A graph is called *k-connected* if there is no set S of at most $k-1$ vertices, such that the graph $G[V(G) - S]$ is disconnected. Such a set S of vertices making G disconnected is called a vertex cut in G. If vertices s and t are then in different components it is called an *s–t vertex cut.* If $s \in U$ and $t \in V - U$ we call $\delta(U)$ an *s–t cut.* Let $U \subseteq V$ then by *contraction* of U we will denote an operation resulting

(22) $$G/U := (V + u - U, E[V - U] \cup \{ux \mid vx \in E \text{ for some } v \in U\}$$

Given $\mathcal{P}$, a family of paths, we will denote

(23) $$\mathrm{ends}_{\mathrm{V}}(\mathcal{P}) := \{s \mid P \in \mathcal{P}, P \text{ is } s\text{–}t \text{ path or } t\text{–}s \text{ path}\}$$

the set of endpoints of those paths. Similarly, we will denote

(24) $$\mathrm{ends}_{\mathrm{E}}(\mathcal{P}) := \{st \mid P \in \mathcal{P}, P \text{ starts or ends with } st\}$$

Clique in G is an induced complete subgraph. A *stable set* is an induced subgraph without edges.

A *directed graph* (digraph) is a pair $G = (V, A)$ where V is a finite set and A a family of ordered pairs from V. Elements of V are called vertices, elements of A arcs.

6. Matchings in graphs

A *matching* in a graph $G = (V, E)$ is a subset M of E such that $e \cap e' = \emptyset$ for all $e, e' \in M, e \neq e'$. A matching is called *perfect* if it covers all vertices, which means $|M| = \frac{1}{2}|V|$. We denote the maximum size of a matching in G by $\nu(G)$. Tutte has proved a characterization of graphs with a prefect matching in [**Tut47**] and later Berge [**Ber58**] extended his result to the famous min–max relation characterizing $\nu(G)$:

THEOREM 2.6 (Tutte–Berge formula). *Let $G = (V, E)$ be a graph. Maximum size of a matching in G is equal to the minimum value of*

(25) $$\tfrac{1}{2}(|V| + |U| - o(G - U))$$

taken over all subsets U of V, where $o(H)$ denotes number of odd–size connected components of H.

Maximum–size matching can be found in polynomial time by algorithm due to Edmonds [**Edm65b**].

6.1. The matching polytope. The *perfect matching polytope* of a graph $G = (V, E)$ is a convex hull of incidence vectors of all perfect matchings in G. We write

(26) $$P_{\text{perfect matching}}(G) = \text{conv.hull}\{\chi_M \mid M \text{ is a perfect matching in } G\}$$

Similarly, we can define a *matching polytope* of a graph $G = (V, E)$ as a convex hull of incidence vectors of all matchings in G. We write

(27) $$P_{\text{matching}}(G) = \text{conv.hull}\{\chi_M \mid M \text{ is a matching in } G\}$$

$P_{\text{perfect matching}}(G)$ and $P_{\text{matching}}(G)$ are polytopes in $\mathbb{R}^E$.

For bipartite graphs, perfect matching polytope can be described by the following system of inequalities:

(28a) $x_e \geq 0$ for each $e \in E$

(28b) $x(\delta(v)) = 1$ for each $v \in V$

as it was shown by Birkhoff in [**Bir46**]. From the perfect matching polytope description for bipartite graphs one can easily derive a system of inequalities describing the matching polytope, which is the following:

(29a) $x_e \geq 0$ for each $e \in E$

(29b) $x(\delta(v)) \leq 1$ for each $v \in V$

First observe that both $P_{\text{matching}}(G)$ and $P_{\text{perfect matching}}(G)$ are in fact polytopes because $P_{\text{perfect matching}}(G) \subseteq P_{\text{matching}}(G) \subseteq [0, 1]^E$. Next we would like to note that the latter description immediately gives us a polynomial algorithm for finding a maximum–cardinality matching (and therefore also a perfect patching, if it exists) for bipartite graphs. Consider the linear program

$$\max\{\mathbf{1}x \mid x \in P_{\text{matching}}(G)\}$$

$P_{\text{matching}}(G)$ is bounded and from the Observation 2.2 we know, that the maximum is attained by some vertex x_0 of $P_{\text{matching}}(G)$. Furthermore, we know that vertices of $P_{\text{matching}}(G)$ are incidence vectors of matchings in G. This specially means that the matching M_0 corresponding to the vertex x_0 has maximal number of edges amongst all matchings of G. It is easy to see that the linear program has polynomially many defining inequalities, therefore we can separate in polynomial time and find the optimum by ellipsoid method. By now we have the size of maximum–cardinality matching. By using the above described algorithm as a black box we can try removing edges from G one by one. Whenever the maximum drops, we have removed the maximum–matching–edge, so we easily obtain edge–set of the maximum matching in polynomial number of queries to the previous algorithm.

For general graphs the inequalities in (28) and (29) are not sufficient. If we take $G = K_3$ a triangle and $x_e = \frac{1}{2}$ for each edge then x satisfies both (28) and (29). It is easy to see that $x \notin P_{\text{perfect matching}}(G)$, since G does not have any perfect matching and therefore $P_{\text{perfect matching}}(G)$ is empty. To see $x \notin P_{\text{matching}}(G)$ consider a linear function $f : \mathbb{R}^E \longrightarrow \mathbb{R}$ defined as $f(y) = \mathbf{1}^T y$. Then $f(x) = \frac{3}{2}$. On the other hand, since $P_{\text{matching}}(G)$ is a polytope this function attains maximum over it in one of its vertices, which are exactly vectors corresponding to the matchings in G. For these vertices x^M, function f computes exactly the size of the corresponding matching M. Maximum size of a matching in K_3 is 1 and so $\max\{f(y) \mid y \in P_{\text{matching}}(G)\} = 1 < \frac{3}{2} = f(x)$, which implies $x \notin P_{\text{matching}}(G)$.

For a general graph G the descriptions of $P_{\text{perfect matching}}(G)$ and $P_{\text{matching}}(G)$ needs additional inequalities. Perfect matching polytope can be described by the following system of inequalities

(30a)	$x_e \geq 0$	for each $e \in E$
(30b)	$x(\delta(v)) = 1$	for each $v \in V$
(30c)	$x(\delta(U)) \geq 1$	for each $U \subseteq V$ with $\lvert U \rvert$ odd

as was shown by Edmonds [**Edm65a**].

Description of the matching polytope for general graph can again be derived from the perfect matching polytope description and consists of the following inequalities:

(31a)	$x_e \geq 0$	for each $e \in E$
(31b)	$x(\delta(v)) \leq 1$	for each $v \in V$
(31c)	$x(E[U]) \leq \lfloor \frac{1}{2} \lvert U \rvert \rfloor$	for each $U \subseteq V$ with $\lvert U \rvert$ odd

As well as in the bipartite case, also these descriptions give us and algorithm for finding the maximum–cardinality matching in polynomial time. The only additional difficulty we have to face is separation over the new condition set of (31c). This can be done in polynomial time but it requires some further knowledge (it involves algorithm for minimal cut in a graph) and we will not show it.

7. *b*–matchings

7.1. *b*–matchings. Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. A *b–matching* is a function $x \in \mathbb{Z}_+^E$ satisfying

$$x(\delta(v)) \leq b(v) \qquad \text{for each } v \in V \tag{32}$$

This can also be rewritten as $Mx \leq b$ where M is the $V \times E$ incidence matrix of G. We remark that the definition allows edges to be used more than once.

In (32) we count multiplicities: if e is a loop at v we count it twice (this is consistent with the definition of $\delta(v)$).

We see that b–matchings are on the one hand generalization of matchings. On the other hand they can be reduced to matching in the following way: we create a graph G_b arising from G by splitting each vertex v into $b(v)$ copies and by replacing every edge uv by $b(u)b(v)$ copies connecting all $b(u)$ copies of u with all $b(v)$ copies of v. Formally, $G_b := (V_b, E_b)$, where

(33a) $$V_b := \{q_{v,i} \mid v \in V, 1 \leq i \leq b(v)\}$$

(33b) $$E_b := \{q_{u,i}q_{v,j} \mid uv \in E, 1 \leq i \leq b(u), 1 \leq j \leq b(v), q_{u,i} \neq q_{v,j}\}$$

The last condition is relevant only if $u = v$ (so the edge is a loop). Hence, b–matchings in G are in correspondence with matchings in G_b and vice versa. The construction above was given by Tutte in [**Tut54**] and it plays a crucial role in proofs of many basic properties of b–matchings and it can be used to adapt algorithm for maximum–cardinality matchings to solve the maximum–cardinality b–matching problem as well.

Similarly as for ordinary matchings, we define the *b–matching polytope* for a graph $G = (V, E)$ as the convex hull of incidence vectors of all b–matchings in G. The inequalities describing this polytope were found by Edmonds [1965] and are:

(34a) $x_e \geq 0$ for $e \in E$

(34b) $x(\delta(v)) \leq b(v)$ for $v \in V$

(34c) $x(E[U]) \leq \lfloor \frac{1}{2} b(U) \rfloor$ for $U \subseteq V$ with $b(U)$ odd

We call x a *perfect b–matching* if it is a b–matching and satisfies (32) with equality (for all v). Then the *perfect b–matching polytope* of a graph $G = (V, E)$ is a convex hull of incidence vectors of all perfect b–matchings. It is described by the following system of inequalities:

(35a) $x_e \geq 0$ for $e \in E$

(35b) $x(\delta(v)) = b(v)$ for $v \in V$

(35c) $x(\delta(U)) \geq 1$ for $U \subseteq V$ with $b(U)$ odd

7.2. Capacitated b–matchings. As a concept of further generalization of b–matchings was studied their capacitated version. Capacitated b–matching problem considers b–matching x satisfying prescribed capacity constraint $x \leq c$. We can again define polytopes for this problem

Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$. The *c–capacitated b–matching polytope* is a convex hull of all b–matchings x satisfying $x \leq c$. Its description follows from the uncapacitated b–matching polytope

(36a) $0 \leq x_e \leq c(e)$ for $e \in E$

(36b) $x(\delta(v)) \leq b(v)$ for $v \in V$

(36c) $x(E[U]) + x(F) \leq \lfloor \frac{1}{2}(b(U) + c(F)) \rfloor$ $U \subseteq V, F \subseteq \delta(U), b(U) + c(F)$ odd

Similarly, the *c–capacitated perfect b–matching polytope* is a convex hull of all perfect b–matchings x of G satisfying $x \leq c$. Its description is

(37a)	$0 \leq x_e \leq c(e)$	for $e \in E$
(37b)	$x(\delta(v)) = b(v)$	for $v \in V$
(37c)	$x(\delta(U) - F) - x(F) \geq 1 - c(F)$	$U \subseteq V, F \subseteq \delta(U), b(U) + c(F)$ odd

This description was announced by Edmonds and Johnson in [**EJ73**].

7.3. Simple b–matchings and b–factors. An interesting case of capacitated b–matchings is when $c = \mathbf{1}$. In such a case x is a 0–1 vector and we can identify matchings with subsets F of $E[G]$ with $\deg_F v \leq b(v)$. We call such b–matchings *simple*. Simple perfect b–matchings are called *b–factors*. Again we can define a *simple b–matching polytope*. Given a graph $G = (V, E)$ and a vector $b \in \mathbb{Z}_+^V$ the *simple b–matching polytope* is a convex hull of incidence vectors of all simple b–matchings in G, which can be described by the following inequalities:

(38a)	$0 \leq x_e \leq 1$	for $e \in E$
(38b)	$x(\delta(v)) \leq b(v)$	for $v \in V$
(38c)	$x(E[U]) + x(F) \leq \lfloor \frac{1}{2}(b(U) + \|F\|) \rfloor$	$U \subseteq V, F \subseteq \delta(U), b(U) + \|F\|$ odd

as a special case of capacitated b–matchings. We define b–factor polytope similarly. Given a graph $G = (V, E)$ and a vector $b \in \mathbb{Z}_+^V$, b–factor polytope is a convex hull of incidence vectors of all b–factors of G. It is described by the following inequalities:

(39a)	$0 \leq x_e \leq 1$	for $e \in E$
(39b)	$x(\delta(v)) = b(v)$	for $v \in V$
(39c)	$x(\delta(U) - F) - x(F) \geq 1 - \|F\|$	$U \subseteq V, F \subseteq \delta(U), b(U) + \|F\|$ odd

8. Paths in graphs

There are many problems concerning disjoint paths in graphs. One well–known is described by the following theorem:

THEOREM 2.7 (Menger's theorem – undirected vertex–disjoint formulation). *Let $G = (V, E)$. G is k–connected if and only if for every $u, v \in V, u \neq v$ there are k disjoint paths connecting u with v.*

Later we will find a *disjoint paths* problem useful. It is the following problem: Given an undirected graph $G = (V, E)$ and pairs of its vertices $(s_1, t_1), \ldots (s_k, t_k)$ find vertex–disjoint paths $P_1, \ldots P_k$ where P_i is an s_i–t_i path for each i.

9. Matroids

Matroids are a powerful tool (not only) in combinatorial optimization. They were motivated by questions in linear algebra, geometry and graph theory. For detailed reading about matroids see Oxley [**Oxl92**] or Schrijver [**Sch03b**].

A pair $M = (S, \mathcal{I})$ is called a *matroid* if S is a finite set and $\mathcal{I}$ is a nonempty collection of subsets of S satisfying

(40a) (i) if $I \in \mathcal{I}$ and $J \subseteq I$ then also $J \in \mathcal{I}$

(40b) (ii) if $I, J \in \mathcal{I}$ and $|I| < |J|$, then $I \cup \{z\} \in \mathcal{I}$ for some $z \in J - I$

Elements of $\mathcal{I}$ are called *independent* sets of the matroid. All other sets $X \subseteq S$ are called *dependent.* Property (i) is called *hereditary property,* (ii) is called *exchange property.*

For a set $U \subseteq S$, a subset B is called a *base* of U if B is its inclusion-wise maximal independent subset. Exchange property gives us immediately:

OBSERVATION 2.8. *Let $(S, \mathcal{I})$ be a matroid and $U \subseteq S$. Then all bases of U have the same size.*

The common size of bases of U is called *rank* of U, denoted by $r(U)$. A set is called a *base*of M if it is a base of S. For $T \subseteq S$, a *span* function is defined as

$$\text{span}(T) := \{s \in S \mid r(T \cup \{s\}) = r(T)\} \tag{41}$$

THEOREM 2.9. *Let $v_1, \ldots, v_m$ be vectors in $\mathbb{R}^n$. Let $S = \{v_1, \ldots, v_m\}$ and let $\mathcal{I}$ consist of all subsets of S formed by linearly independent vectors. Then $(S, \mathcal{I})$ is a matroid.*

The above defined matroid is called a *linear matroid.* Note that for a linear matroid, bases correspond to a linear algebraic bases, rank to the dimension of a generated space and span to the set of vectors of S in the generated space.

Sometimes infinite matroids are also considered — for example for matroid matchings defined later. An *infinite matroid* M is a pair $(S, \mathcal{I})$ where S is an infinite set and $\mathcal{I}$ collection of its subsets satisfying

(42a) (i) if $I \in \mathcal{I}$ and $J \subseteq I$ then also $J \in \mathcal{I}$

(42b) (ii) if $I \subseteq S$ and all finite subsets of I belong to $\mathcal{I}$, then I belongs to $\mathcal{I}$

(42c) (iii) if $I, J \in \mathcal{I}$, $|I| < |J| < \infty$, then $I \cup \{z\} \in \mathcal{I}$ for some $z \in J - I$

A straightforward infinite extension of linear matroids are *linear spaces* $\mathbb{R}^n$ where $\mathcal{I}$ consists of all linearly independent subsets of $\mathbb{R}^n$.

Let $(S, \mathcal{I})$ be a matroid with rank function r and span function span. Let E be a finite collection of unordered pairs from S such that each pair is an independent set. For $F \subseteq E$ define

$$\text{span}(F) := \text{span}(\cup F) \tag{43}$$

$$r(F) := r(\text{span}(F)) \tag{44}$$

A subset M of E is called a *matroid matching* or *matching*, if

$$r(M) = 2|M| \tag{45}$$

So M is a matching if and only if it consists of disjoint pairs and union of those pairs is independent.

THEOREM 2.10. *Given a set E of pairs of vectors in a linear space L, a maximum-size matching can be found in strongly polynomial time.*

10. P and NP

One of the major fields in computer science studies complexity issues and helps us believe, that some problems are hard to solve. An exhaustive reading on this topic from the basics can be found in Papadimitriou [**Pap94**].

Let $\mathcal{L}$ be a language over alphabet A. A *decision problem* for this language is to decide for a given word $w \in A^*$ whether $w \in \mathcal{L}$. P is a class of languages, for which there exists a deterministic Turing machine deciding whether $w \in \mathcal{L}$ in polynomially (in the length of w) bounded number of steps. NP is a class of languages, for which there exists a non–deterministic Turing machine deciding whether $w \in \mathcal{L}$ in polynomially (in the length of w) bounded number of steps. Obviously $P \subseteq NP$. Roughly speaking, P can be considered a class of problems for which there exists a polynomial algorithm and NP a class of problems for which there is a certificate for a positive answer, which can be verified in polynomial time. Problem $\mathcal{L}$ is called *NP–hard* if for every problem $\mathcal{L}_0$ in NP there is a polynomial–time algorithm transforming any instance I_0 of $\mathcal{L}_0$ to an instance I of $\mathcal{L}$ such that $I \in \mathcal{L}$ if and only if $I_0 \in \mathcal{L}_0$ (we say $\mathcal{L}_0$ reduces to $\mathcal{L}$). As soon as we have any NP–hard problem $\mathcal{L}_{NP}$ it suffices to find a reduction of $\mathcal{L}_{NP}$ to $\mathcal{L}$ in order to show that $\mathcal{L}$ is NP–hard. A problem is called NP–complete if it is in NP and is NP–hard.

All NP–hardness result for a problem $\mathcal{L}$ give us hope for a polynomial algorithm for deciding $\mathcal{L}$ only under assumption $P = NP$, which is widely believed to be untrue.

11. Some NP–hard problems

For further reading we will need to know NP–hardness results about the following two problems. The first one is a decision variant (since we have defined NP only for decision problems) of the disjoint paths problem mentioned in Section 8.

THEOREM 2.11. *Let $\mathcal{DP}$ be a decision problem defined as follows:*

Given a graph $G = (V, E)$, integer $k \in \mathbb{N}$ and k pairs of vertices $(s_1, t_1), \dots, (s_k, t_k)$, decide whether there exist disjoint paths $(P_i)_{i=1}^k$ such that P_i connects s_i with t_i.

Problem $\mathcal{DP}$ is NP–hard.

Another NP–hard problem we will be dealing with later is:

THEOREM 2.12. *The following problem is NP-hard:*
Let $G = (V, E)$ be a graph. Determine whether there is a path P in G containing all vertices of G.

Such a path is called *Hamiltonian path.* The problem is called *Hamiltonian path problem.*

12. Some polynomially solvable problems

We will also need to know that there are polynomial algorithms for the min-cut problem:

THEOREM 2.13. *Let $G = (V, E)$ be a graph and $c \in \mathbb{R}^E$ be called a capacity function. Let $S, T \subseteq V, S \cap T = \emptyset$. Then the minimal-capacity cut $\delta(U)$ such that $S \subseteq U, T \subseteq V - U$ can be found in polynomial time.*

and the min–odd–cut problem:

THEOREM 2.14. *Let $G = (V, E)$ be a graph and $c \in \mathbb{R}^E$. Let $s, t \in V, s \neq t$. Then there is a polynomial-time algorithm for finding a minimum-capacity odd (in number of edges) s–t cut.*

CHAPTER 3

Path packings

In the introductory chapter we looked at matchings and their generalizations — the b–matchings.

Now we will go over another related problem of finding a maximum number of disjoint paths whose sets are two different vertices in a given set T of vertices (compared to matchings where we are looking for a maximum number of disjoint edges) — the T–paths. Fundamental theorems of Mader imply min–max relations for this.

Let $G = (V, E)$ be a graph and $T \subseteq V$. A path P in G is called a *T-path* if its ends are distinct vertices in T and no internal vertex of P belongs to T.

Mader [**Mad78b**] gave a min–max formula for the maximum number of internally vertex–disjoint T–paths. It is a generalization of the undirected vertex–disjoint version of Menger's theorem (for $|T| = 2$) and also of Tutte–Berge formula (by adding a copy v' to each vertex v of the original graph G and connecting it by an edge $v'v$; if we take T as the set of all those copies, the maximum number of internally vertex–disjoint T–paths in the new graph is equal to the maximum size of a matching in the original graph).

There are different paths–packing problems and their min – max characterizations.

1. Fully vertex–disjoint T–paths

The simplest one is the fully vertex–disjoint version.

THEOREM 3.1 (Gallai's disjoint T–paths theorem). *Let $G = (V, E)$ be an undirected graph and let $T \subseteq V$. The maximum number of (fully) vertex–disjoint T–paths in G is equal to the minimum value of*

$$|U| + \sum_K \lfloor \tfrac{1}{2}|K \cap T| \rfloor \tag{46}$$

taken over all $U \subseteq V$, where K ranges over all components of $G - U$.

This problem is closely related to the ordinary maximum–size matching problem and can directly reduced to it – as Gallai [**Gal61**] showed in his proof.

PROOF. The maximum is less or equal to the minimum, since for every $U \subseteq V$ each T–path either intersects U (and hence, uses at least one of its vertices) or has both endpoints in a single $K \cap T$ for some component K of $G - U$.

To see that the maximum is at least the minimum, let μ be the minimal value of (46). Let $\hat{G} = (\hat{V}, \hat{E})$ be a graph constructed from G by adding a disjoint copy H of $G - T$ and making copy $v' \in H$ of each vertex $v \in V - T$ adjacent to v and all its neighbours in G. Formally

$$\begin{aligned} \hat{V} =& V \cup \{v' \mid v \in V - T\} \\ \hat{E} =& E \cup \{u'v' \mid uv \in E, u, v \in V - T\} \cup \\ & \{v'u \mid v \in V - T, u \in N(V) \cup \{v\}\} \end{aligned} \tag{47}$$

From Tutte-Berge formula we will derive that $\hat{G}$ has a matching of size $\mu + |V - T|$. To show it we must prove that for any $\hat{U} \subseteq \hat{V}$ the following inequality holds:

$$|\hat{U}| + \sum_{\hat{K}} \lfloor \tfrac{1}{2} |\hat{K}| \rfloor \geq \mu + |V - T| \tag{48}$$

where $\hat{K}$ ranges over the components of $\hat{G} - \hat{U}$. If for some $v \in V - T$ exactly one of v, v' belongs to $\hat{U}$ we can delete it from $\hat{U}$ while not increasing the left–hand side of (48). Value $|\hat{U}|$ drops by one and the sum does not increase by more than one, since just one component that previously contained v or v' will change (by adding the other vertex from this pair).

So we assume that for each $v \in V - T$ either $v, v' \in \hat{U}$ or $v, v' \notin \hat{U}$. Define $U := \hat{U} \cap V$. Then each component K of $G - U$ is equal to $\hat{K} \cap V$ for some component $\hat{K}$ of $\hat{G} - \hat{U}$ (v, v' always have the same set of neighbours). Additionally, the rounding in the sum can be caused just by presence of a vertex from T in the component because all other vertices are paired with their copies in the components. Hence

$$\hat{U} + \sum_{\hat{K}} \lfloor \tfrac{1}{2} |\hat{K}| \rfloor = |U| + \sum_{K} \lfloor \tfrac{1}{2} |K \cap T| \rfloor + |V - T| \geq^{(46)} \mu + |V - T| \tag{49}$$

where K ranges over all components of $G - U$. This gives us (48) and therefore a matching M in $\hat{G}$ of size $\mu + |V - T|$. Consider the matching $N = \{vv' \mid v \in V - T\}$ in $\hat{G}$. Then union of M and N has at least μ components with more edges in M than in N since $|M| = \mu + |V - T| = \mu + |N|$. Each such component gives us a path connecting two vertices in T and after contracting edges of N also a T–path in G. So the set of those components gives us μ desired T–paths. □

2. Disjoint $\mathcal{S}$-paths

The strongest version of min – max relation concerning disjoint T-paths speaks about $\mathcal{S}$-paths. Let $G = (V, E)$ be a graph, let $\mathcal{S}$ be a collection of disjoint subsets of V. A path in G is called an $\mathcal{S}$-path if it connects vertices $s_1 \in S_1, s_2 \in S_2$ from two different sets $S_1, S_2 \in \mathcal{S}$. We denote $T := \cup\mathcal{S}$.

THEOREM 3.2 (Mader's disjoint $\mathcal{S}$-paths theorem). *The maximum number of disjoint $\mathcal{S}$-paths in G is equal to the minimum value of*

$$|U_0| + \sum_{i=1}^{n} \lfloor \tfrac{1}{2}|B_i| \rfloor \tag{50}$$

taken over all partitions $U_0, U_1, \ldots, U_n$ of V such that each $\mathcal{S}$-paths intersects U_0 or traverses some edge spanned by some U_i. B_i then denotes the set of vertices in U_i that belong to T or have a neighbour in $V - (U_0 \cup U_i)$. The value of the expression above for the partition $U_0, U_1, \ldots, U_n$ is called value of the partition.

It can be easily seen that Theorem 3.2 immediately implies Theorem 3.1 when we take $\mathcal{S} = \{\{s\} \mid s \in T\}$.

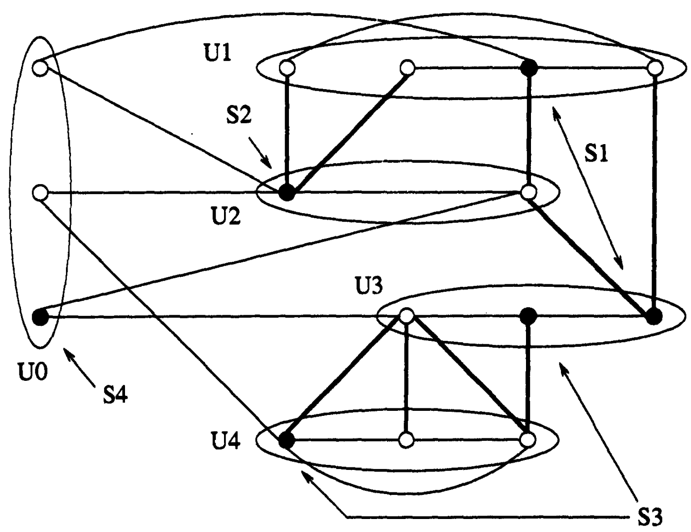

FIGURE 1. Dual partition for $\mathcal{S}$-paths

The condition "such that each $\mathcal{S}$ path intersects U_0 or traverses some edge spanned by some U_i" might sound counter-intuitive. We explain briefly what it means. The above stated says that the graph $G' := G - U_0 - \cup_{i=1}^{n} E[U_i]$ has no two vertices s, t, such that $s \in S_i$, $t \in S_j$ and $i \neq j$, in the same component. We can see an example of such a partition in Figure 1 (vertices of T are black, edges of G' are in bold).

A short proof using Gallai's theorem was given by Schrijver in [**Sch01**]:

PROOF. Let μ be the the minimum value of (50). Trivially, the maximum number of disjoint $\mathcal{S}$-paths is at most μ, since each $\mathcal{S}$-path P either intersects U_0 in at least one vertex or traverses an edge of some U_i and therefore, it contains at least two vertices of B_i.

Fixing V, we choose a counterexample $E, \mathcal{S}$ minimizing

$$|E| - |\{\{i,j\} \mid i,j \in V, \exists I, J \in \mathcal{S} : i \in I, j \in J, I \neq J\}|. \tag{51}$$

The base case where $|X| = 1$ for each $X \in \mathcal{S}$ follows from Gallai's theorem (Theorem 3.1): Every $\mathcal{S}$–path also is a T–path and vice-versa. Therefore the maxima in both theorems are equal. If we take a minimizing partition $U, K = \{K_1, \ldots K_r\}$ from the Gallai's theorem and define $U_0 := U, U_1 := K_1, \ldots, U_r := K_r$, the resulting partition $U_0, \ldots U_n$ is a valid partition in Mader's theorem (because each $\mathcal{S}$–path either enters U or is fully contained in one component K_i and therefore traverses at least one edge spanned by K_i). Hence, the maximum number of disjoint $\mathcal{S}$–paths is at least μ, proving our special case.

So $|X| \geq 2$ for some $X \in \mathcal{S}$. Then X is a stable set, since deleting an edge e spanned by X increases the value of (51), but does not change the minimum and maximum value in Mader's theorem (as no $\mathcal{S}$–path is by definition allowed to use e and deleting e does not change any set B_i). Choose an $s \in X$ and define

$$\mathcal{S}' := (\mathcal{S} - \{X\}) + \{X - \{s\}\} + \{s\} \tag{52}$$

Replacing $\mathcal{S}$ by $\mathcal{S}'$ decreases (51), but does not decrease the minimum in Mader's theorem (as every $\mathcal{S}$–path is an $\mathcal{S}'$–path and $\cup\mathcal{S}' = T$, therefore every feasible partition for $\mathcal{S}'$ in Mader's theorem is a valid partition for $\mathcal{S}$ as well and B_i are equal in both cases). Hence there is a family $\mathcal{P}$ of μ disjoint $\mathcal{S}'$–paths.

Then there is a path P_0 connecting s with another vertex of X (otherwise $\mathcal{P}$ is a family of μ disjoint $\mathcal{S}$–paths, contradicting the fact that $G, \mathcal{S}$ is a counterexample) and all other paths in $\mathcal{P}$ are $\mathcal{S}$–paths. Let u be an internal vertex of P_0 (it exists because X is a stable set). Define

$$\mathcal{S}'' := (\mathcal{S} - \{X\}) + \{X + \{u\}\} \tag{53}$$

Replacing $\mathcal{S}$ by $\mathcal{S}''$ again decreases (51) but does not decrease the minimum in Mader's theorem (by the same argument as in the previous case, since every $\mathcal{S}$–paths is an $\mathcal{S}''$–path as well and $T \subseteq \cup\mathcal{S}''$). So there is a family $\mathcal{Q}$ of μ disjoint $\mathcal{S}''$–paths. Choose $\mathcal{Q}$, such that

$$E[\cup\mathcal{Q}] - E[\cup\mathcal{P}] \tag{54}$$

is minimized.

Necessarily, u is an endpoint of some path $Q_0 \in \mathcal{Q}$ (otherwise $\mathcal{Q}$ forms μ disjoint $\mathcal{S}$–paths, again contradicting the fact that $G, \mathcal{S}$ is a counterexample) and all other paths in $\mathcal{Q}$ are $\mathcal{S}$–paths (see Figure ??). As $|\mathcal{P}| = |\mathcal{Q}|$ and u is an endpoint of Q_0 while it is not an endpoint of any path in $\mathcal{P}$, there is some endpoint r of some path $P \in \mathcal{P}$, which is not an endpoint of any path in $\mathcal{Q}$.

Then P intersects some path in $\mathcal{Q}$ (otherwise $\mathcal{Q} - Q_0 + P$ is a family of μ disjoint $\mathcal{S}$–paths). Starting in r and following the path P, denote the first encountered vertex of $\mathcal{Q}$ by w belonging to a path $Q \in \mathcal{Q}$. Then Q is split by w into two parts Q', Q'' (possibly of zero–length). Denote the part of P connecting r and w by P'.

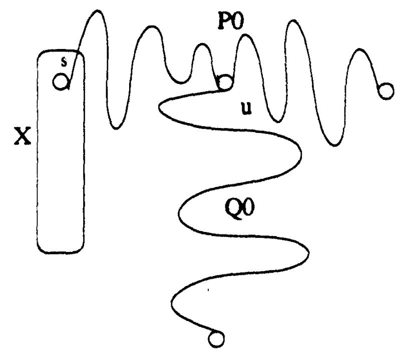

FIGURE 2. Situation in the proof: P_0 and Q_0

If neither of the parts Q', Q'' is fully contained in the path P, then one endpoint of Q belongs to a different set from $\mathcal{S}''$ than r does. Without loss of generality it is an endpoint of Q'. Then joining the path Q' with the path P' yields an $\mathcal{S}''$–path disjoint form all the other paths in $\mathcal{Q}$. This contradicts the assumption on minimality of (54). So we can assume that $EQ' \subseteq EP$.

If $P = P_0$ then we have $Q = Q_0$ (since $EQ' \subseteq EP$ and w is the first point of $\mathcal{Q}$ on the way from r, the path Q must start in u). Let Q_P be a path obtained by replacing the part Q' of Q by P'. Then the family $\mathcal{Q} - Q + Q_P$ consists of μ disjoint $\mathcal{S}$ paths, which again leads to a contradiction.

Otherwise $P \neq P_0$. Then P is a $\mathcal{S}''$–path as well and it is disjoint from all the paths in $\mathcal{Q}$ apart from Q (as Q' must be the same path as $P - P'$). Then the family $\mathcal{Q} - Q + P$ contradicts the minimality assumption on (54) and finishes the proof by showing that there is no such counterexample to Mader's theorem. □

What we have just proven is closely related to the following:

3. Internally vertex–disjoint T–paths

The original theorem of Mader was stated in terms of internally vertex–disjoint T–paths (endpoints can be shared by more paths but internal vertices have to be disjoint) instead of fully vertex–disjoint $\mathcal{S}$–paths and can be derived from the Theorem 3.2.

For a graph G define $B_G(U)$ set of vertices in U having a neighbour in G that is not in U.

COROLLARY 3.3 (Mader's internally vertex–disjoint T–paths theorem). *Let $G = (V, E)$ be a graph and let T be a stable set of V. Then the maximum number of internally vertex–disjoint T–paths in G is equal to the minimum value of*

$$|U_0| + \sum_{i=1}^{n} \lfloor \tfrac{1}{2} |B_{G-U_0}(U_i)| \rfloor \tag{55}$$

where $U_0, U_1, \ldots, U_n$ partition $V - T$ such that each T–path intersects U_0 or traverses some edge spanned by some U_i.

PROOF. The maximum is less or equal to the minimum since each path not intersecting U_0 traverses at least two vertices in some U_i hence it traverses at least two vertices of U_i that have a neighbour in $V - (U_0 \cup U_i)$.

To prove equality we either have two vertices of T having a common neighbour v or not. In the former case we can delete v from G. Then the maximum drops by 1 and the minimum also drops by 1 since v must be in U_0. So we can apply induction to show the equality. Otherwise there are no two vertices of T having common neighbour and we can apply Theorem 3.2 with $G - T$ and $\mathcal{S} = \{N(s) \mid s \in T\}$. Therefore the maximum number of T–paths in G is equal to the maximum number of $\mathcal{S}$–paths in $G - T$, which is equal to the minimum value of

$$|U_0|+\sum_{i=1}^{n}\lfloor\tfrac{1}{2}|B_i|\rfloor = |U_0|+\sum_{i=1}^{n}\lfloor\tfrac{1}{2}|B_{G-U_0}(U_i)\cup(U_i\cap T)|\rfloor = |U_0|+\sum_{i=1}^{n}\lfloor\tfrac{1}{2}|B_{G-U_0}(U_i)|\rfloor$$

taken over $U_0, U_1, \ldots, U_n$ partitions of $V(G-T) = V - T$ such that each $\mathcal{S}$–path intersects U_0 or traverses an edge spanned by some U_i. Every such partition also satisfies condition that every T–path in G either intersects U_0 or traverses some edge spanned by some U_i, which proves the corollary. □

4. Edge disjoint T–paths

Similar result for edge–disjoint T–paths packings is also known and is implied by Corollary 3.3 as Mader has shown in [**Mad78a**].

THEOREM 3.4 (Mader's edge disjoint T–paths theorem). *Let $G = (V, E)$ be a graph and let $T \subseteq V$. Then the maximum number of edge–disjoint T-paths in G is equal to the minimum value of*

$$\tfrac{1}{2}\Big(\sum_{s\in T} d_E(X_s) - \kappa\Big) \tag{56}$$

where the X_s are disjoint sets with $s \in X_s$ for $s \in T$, and where κ is the number of components K of a graph $G - \cup_{s\in T} X_s$ with $d_E(K)$ odd.

5. Mader matroid

Mader's disjoint $\mathcal{S}$–paths define a matroid. Let $S = \cup\mathcal{S}$ and let $\mathcal{I}$ consist of all I such that $I \subseteq \mathrm{ends}_V(\mathcal{P})$ for some family $\mathcal{P}$ of disjoint $\mathcal{S}$–paths. Then

THEOREM 3.5. *The pair $M := (S, \mathcal{I})$ defined above is a matroid.*

The hereditary property is obvious. The exchange property can be extracted from the proof of Mader's $\mathcal{S}$–path theorem from [**Sch01**], the proof can be found in [**Sch03c**]. The above defined matroids are called *Mader matroids.*

There was an open question on the relation of Mader matroids and another known class of matroids – so called gammoids. This question was recently answered by Pap in [**Pap06**] resulting that every Mader matroid is a gammoid and vice versa.

6. Non-zero T-paths

Recently there were few further extensions of vertex-disjoint T-paths packing. First of them is due to Chudnovsky, Geelen, Gerards, Goddyn, Lohman and Seymour [**CGG⁺04**], who have introduced a new, more general concept of non-zero T-paths packing in so-called group-labeled graphs.

Let Γ be a group (we will use additive notation for Γ even though it does not have to be abelian), $G = (V, E)$ be an oriented graph and $\gamma \in \Gamma^E$. A triple $\mathcal{G} = (G, \Gamma, \gamma)$ will be called a *group-labeled graph.* Let $e = (u, v) \in E$. We will denote $\gamma(e, u) = -\gamma_e$ and $\gamma(e, v) = \gamma_e$. Now for a path $P = (v_0, e_1, v_1, e_2, \ldots, e_k, v_k)$ in G the *weight* of P denoted as $\gamma(P)$ will be $\gamma(P) = \sum_{i=1}^{k} \gamma(e_i, v_i)$.

Let $\mathcal{G} = (G, \Gamma, \gamma)$ be a group-labeled graph and $T \subseteq V(G)$ be a set of vertices of $\mathcal{G}$. We will call a path or a circuit P non-zero if $\gamma(P) \neq 0_\Gamma$. A set of edges $F \subseteq E(G)$ is *T-balanced* if F contains no non-zero path or circuit. Denote $V(F)$ a set of all vertices incident to some edge in F. It is easy to see that F is T-balanced if and only if all vertices in $V(F)$ can be assigned labels γ_v from Γ according to the following rule

(57a) $\quad \gamma_v = 0_\Gamma \qquad v \in V(F) \cap T$

(57b) $\quad \gamma_v = \gamma_u + \gamma(e, u) \qquad e \in F, e = (v, u)$ or $e = (u, v)$

Authors have proved a min-max relation for a problem of maximum number of (fully) vertex-disjoint non-zero T-paths.

THEOREM 3.6 (Chudnovsky et. al. '04). *Let $\mathcal{G} = (G, \Gamma, \gamma)$ be a group-labeled graph and let $T \subseteq V(G)$. Then the maximum number of vertex-disjoint non-zero T-paths in G is equal to the minimum value of*

$$|X| + \sum_{H \in C_{X,F}} \lfloor \tfrac{1}{2} |(T \cup V(F)) \cap V(H)| \rfloor \tag{58}$$

taken over all T-balanced sets $F \subseteq E(G)$ and all sets $X \subseteq V(G)$ where $C_{X,F}$ denotes the set of components of $G - X - F$.

Proof of this theorem can be found in Chudnovsky et. al. [**CCG04**]. It can be seen that Gallai's vertex-disjoint T-paths packing and Mader's vertex-disjoint $\mathcal{S}$-paths packing are a special case of non-zero T-paths. In the following paragraph, we explain the relation:

T-paths: we want every T-path to be feasible. Consider $\Gamma = \mathbb{Z}_{|T|}$. Let $T = \{t_1, t_2, \ldots, t_{|T|}\}$. For $e = (u, v) \in E(G)$ we define γ as follows:

$$\gamma_{(u,v)} = \begin{cases} i - j & u = t_i, v = t_j \\ i & u = t_i, v \in V(G) - T \\ -i & u \in V(G) - T, v = t_i \\ 0 & \text{otherwise.} \end{cases}$$

In such a configuration every path connecting vertices t_i, t_j has its weight equal to $i - j \neq 0$.

$\mathcal{S}$-paths: we want every $\mathcal{S}$–path to be feasible and every path connecting two vertices in a single $S \in \mathcal{S}$ to be infeasible. Consider $\Gamma = \mathbb{Z}_{|\mathcal{S}|}$. Let $\mathcal{S} = \{S_1, S_2, \ldots, S_{|\mathcal{S}|}\}$ and let $T = \cup\mathcal{S}$. For edges $e = (u, v)$ we define γ as follows:

$$\gamma_{(u,v)} = \begin{cases} i - j & u \in S_i, v \in S_j \\ i & u \in S_i, v \in V(G) - T \\ -i & u \in V(G) - T, v \in S_i \\ 0 & \text{otherwise.} \end{cases}$$

In such a configuration every path connecting vertices in S_i and S_j has a weight equal to $i - j$. This is not 0 if and only if $i \neq j$.

7. Non–returning T–paths

The previous concept was further extended to a more general concept of non–returning T–paths in permutation–labeled graphs by Pap [**Pap05b**].

Let $G = (V, E)$ be an oriented graph and $T \subseteq E$ a set of *terminals*. Let Ω be an arbitrary *set of potentials* and let $\omega : T \longrightarrow \Omega$ define a *potential of origin* for the terminals. Let $\pi : E \longrightarrow S(\Omega)$ where $S(\Omega)$ denotes the set of all permutations on Ω. We will call a *permutation–labeled graph* such $\mathcal{G} = \{G, T, \Omega, \omega, \pi\}$. A *$T$–path* in $\mathcal{G}$ is then a T–path in the underlying undirected graph. For an arc $e = (a, b)$ we denote $\pi(e, a) = \pi^{-1}(e)$ and $\pi(e, b) = \pi(e)$. For a T–path $P = (v_0, e_1, v_1, \ldots, e_k, v_k)$ let $\pi(P) = \pi(e_1, v_1) \circ \pi(e_2, v_2) \circ \ldots \circ \pi(e_k, v_k)$. P is called non–returning if $\pi(P)(\omega(v_0)) \neq \omega(v_k)$ — in other words, the path P does not map potentials of origin to each other.

Let $\mathcal{G} = (G, T, \Omega, \omega, \pi)$ be a permutation–labeled graph and $F \subseteq E(G)$. Denote $V(F)$ a set of vertices incident to some edge of F. Let $T' = T \cup V(F)$. F is called *T–balanced* if ω can be extended to $\omega' : A' \longrightarrow \Omega$ such that $\pi(ab)(\omega'(a)) = \omega'(b)$ for each arc $ab = e \in F$.

Pap has shown a min–max relation for the maximum number of (fully) vertex–disjoint non–returning T–paths in a permutation–labeled graph.

THEOREM 3.7 (Pap '05). *Let $\mathcal{G} = (G, T, \Omega, \omega, \pi)$ be a permutation–labeled graph. Maximum number of vertex–disjoint non–returning T–paths in G is equal to the minimum value of*

$$\nu'(G - F, T \cup V(F)) \tag{59}$$

taken over all T–balanced edge sets F. $\nu'(G, T)$ denotes the maximum number of vertex–disjoint T–paths in G.

The proof of this theorem can be found in [**Pap05b**].
This theorem is a generalization of Theorem 3.6. It can be easily seen by taking Ω

equal to the underlying set of the group Γ. Then define $\omega(t) = 0_\Gamma$ for each $t \in T$. Let $\pi(e)$ be functions on Ω defined by $\pi(e)(x) = x + \gamma(e)$. It is a permutation since the '+' operation has inverse.

CHAPTER 4

Algorithmic aspects of paths packings

1. Gallai's paths and matchings

In the proof of Gallai's theorem the problem of finding a maximum fully vertex–disjoint T–paths packing was reduced to problem of finding a maximum–size matching in an auxiliary graph.

We can easily find the maximum set of paths by the following algorithm. At first construct an auxiliary graph $\hat{G} = (\hat{V}, \hat{E})$ according to the rule (47). Now we find a maximum–size matching M in $\hat{G}$. Then let $N = \{\{v, v'\} \mid v \in V - T\}$ be a trivial matching of size $|V - T|$. Take all the components of $(V, M \cup N)$ having more edges in M than in N and contract edges of N to obtain a family $\mathcal{P}$ of disjoint T–paths in G. From the proof of Gallai's theorem we have that such $\mathcal{P}$ is a maximum–size family.

2. Matroid matching

Lovász [**Lov80**] showed that Mader's theorem can be derived from matroid matching theory, which yields a polynomial–time algorithm for finding a maximum packing of $\mathcal{S}$–paths by matroid–matching algorithm for linear spaces [**Lov81**]. The following text is restricted just to derive a polynomial–time solvability of the problem as in [**Sch03c**].

THEOREM 4.1. *There is a strongly polynomial algorithm for finding maximum number of disjoint $\mathcal{S}$-paths in a graph.*

PROOF. Let $G = (V, E)$ be a graph and $\mathcal{S} = \{S_1, \dots, S_k\}$ be a set of disjoint subsets of V. Let $T = \cup\mathcal{S} = S_1 \cup \dots \cup S_k$. We can assume that each S_i is a stable set (because no edge uv where $u, v \in S_i$ could ever contribute to an $\mathcal{S}$–path).

Consider a linear space $(\mathbb{R}^2)^V$ which can be seen as a space of functions $x : V \longrightarrow \mathbb{R}^2$ but also as a space of vectors in $\mathbb{R}^{2\times|V|}$. For each edge $e = uv \in E(G)$, let L_e be the linear subspace of $(\mathbb{R}^2)^V$ given by

$$(60) \qquad L_e = \{x \in (\mathbb{R}^2)^V \mid x(w) = 0 \text{ for each } w \notin \{u, v\} \text{ and } x(u) = -x(v)\}$$

Alternatively it can be seen as a space of vectors x having all components equal to zero except for x_u, x_v: $x_u \in \mathbb{R}^2, x_v = -x_u$. It is obvious that $\dim L_e = 2$.

Choose distinct 1–dimensional subspaces $l_1, \dots, l_k$ of $\mathbb{R}^2$ and for each vertex $v \in V$ define $L_v = l_i$ if $v \in S_i$ for some i or $\{\mathbf{0}\}$ otherwise. Define

$$(61) \qquad \mathcal{Q} = \{x \in (\mathbb{R}^2)^V \mid \forall v \in V : x(v) \in L_v\}$$

Let $\mathcal{E}$ be the collection of all (for all $e \in E$) subspaces L_e/Q of $(\mathbb{R}^2)^V/Q$. For each edge $e \in E$ the dimension $\dim(L_e/Q) = \dim(L_e) - \dim(L_e \cap Q)$. Since $\dim(L_e) = 2$ and no two vertices of an edge are in the same S_i so $L_e \cap Q = \{\mathbf{0}\}$, we have $\dim(L_e/Q) = 2$.

For any $F \subseteq E$, let $\mathcal{L}_F$ denote the following collection of spaces in $\mathcal{E}$

$$\mathcal{L}_F = \{L_e/Q \mid e \in F\} \tag{62}$$

We show for each $F \subseteq E$

LEMMA 4.2. *$\mathcal{L}_F$ is a matching if and only if F is a forest such that each component of (V, F) has at most two vertices in common with T and at most one with each S_i*

Let $X = \sum_{e \in F} L_e$. Then X consists of all $x : V \to \mathbb{R}^2$ with $x(K) = 0$ for each component K of (V, F) since X is a linear hull of all vectors in all L_e, which means $x \in X$ if and only if x can be expressed as $x = \sum_{e \in F} \alpha_e y_e$ where α_e are real weights and $y_e \in L_e$. By such a sum one can just obtain x such that $x(K) = 0$ for each component K of (V, F) since adding any multiple of any y_e does not change the sum over the component. Conversely if x has zero sum over each component of (V, F) it can be easily expressed in a given form by taking spanning tree of the component and then assigning weights to the edges greedily from the leaves in order to satisfy the leaf (then remove it and continue recursively).

So $\dim(X) = (2|V| - \kappa)$ where κ is number of components of (V, F). Then $\dim(X \cap Q) = 0$ if and only if each component of (V, F) has at most two vertices in common with T and at most one vertex in common with every S_i (we have non–empty intersection if and only if there is a set of linearly dependent non–zero vectors $x(u)$ for $u \in K \cap T$, which happens if and only if there are at least three non–zero vectors or two colinear vectors $x(u)$ in one $K \cap T$). Now

$$\dim(\mathcal{L}_F) = \dim(X/Q) = \dim(X) - \dim(X \cap Q) \leq \dim(X) \leq 2|F| \tag{63}$$

Hence $\mathcal{L}_F$ has dimension $2|F|$ and therefore is a matching if and only if $\dim(X \cap Q) = 0$ and $\dim(X) = 2|F|$. By the previous we have proved Lemma 4.2

Lemma 4.2 then implies the following relation to $\mathcal{S}$–paths.

LEMMA 4.3. *If G is connected, the maximum number of disjoint $\mathcal{S}$–paths is equal to $\nu(\mathcal{E}) - |V| + |T|$.*

To see this let t be the maximum number of disjoint $\mathcal{S}$–paths, let $\mathcal{P}$ be the family of t disjoint $\mathcal{S}$–paths and let F' be set of edges contained in these paths. Extend F' to F such that each component in (V, F) either contains a unique path in $\mathcal{P}$ or a unique vertex in T. Then F satisfies condition given in Lemma 4.2, and $|F| = |V| - |T| + t$ since there are $|T| - t$ components. So $\mathcal{L}_F$ forms a matching of size $t + |V| - |T|$, and hence $\nu(\mathcal{E}) \geq t + |V| - |T|$. Conversely, let $\mathcal{F} \subseteq \mathcal{E}$ be a matching of size $\nu(\mathcal{E})$. Then $\mathcal{F}$ is some forest $(V, F^\star)$ satisfying the condition given in Lemma 4.2. Let $t^\star$ be the number of components of $(V, F^\star)$ intersecting

T twice (and thus containing a unique $\mathcal{S}$–path). Then by deleting $t^\star$ edges from $F^\star$, we obtain a forest such that each component intersects T at most once. So $|F^\star| - t^\star \leq |V| - |T|$ since the resulting forest has T components, and hence

$$t^\star \geq |F^\star| - |V| + |T| = \nu(\mathcal{E}) - |V| + |T|$$

This shows Lemma 4.3. Together with Theorem 2.10 it gives us strongly–polynomial algorithm for finding maximum number of disjoint $\mathcal{S}$ paths in G. □

Such an algorithm also gives us a polynomial algorithm for finding a family attaining this maximum since we can always do the following:

At first we determine the actual maximum $\mathcal{S}$–paths packing size t. Now for each edge $e \in E$ we determine the maximum $\mathcal{S}$–paths packing size t' in $G - e$. If $t' = t$ the edge is not necessary for some optimal solution. Thus we delete it and continue with $G^{NEW} := G - e$. Otherwise the edge is needed for all optimal solution and we have to keep it. The graph we have in the end after proceeding with all edges is exactly a union of $\mathcal{S}$–paths from the optimal solution. Those paths can be easily identified therefore we determine the optimal family $\mathcal{P}$ in polynomial time.

3. Edge–disjoint T–paths and linear programming

Keijsper, Pendavingh, Stougie proved in **[KPS06]** from Mader's edge–disjoint T–paths theorem that one can find maximum number of edge–disjoint T–paths in polynomial time by ellipsoid method.

To see this, let $G = (V, E)$ be a graph and let $T \subseteq V$. Consider the polyhedron $P \in \mathbb{R}^E$ given by x satisfying

(64a)	$0 \leq x_e \leq 1$	for each $e \in E$
(64b)	$x(\delta(U)) \leq \lvert\delta(U)\rvert - 1$	for each $U \subseteq V - T$ with $\lvert\delta(U)\rvert$ odd
(64c)	$x(\delta(s)) \leq x(\delta(X))$	for each $s \in T, X \subseteq V$ with $X \cap T = \{s\}$

All of these conditions can be tested in polynomial time. The conditions (64a) can be checked trivially one by one. To test (64b) let G' be a graph obtained from G by contracting T to a single vertex t. Let c be a capacity function given by $c_e := 1 - x_e$. Then (64b) is satisfied if and only if the minimum capacity of an odd t cut in G' is at least 1. That can be tested in polynomial time due to Theorem 2.14. Finally, testing (64c) can also be done by finding a cut separating s and $T - \{s\}$ of minimum capacity taking x as a capacity function for every $s \in T$ due to Theorem 2.13.

Thus one can optimize any linear function over P by the ellipsoid method in polynomial time.

LEMMA 4.4. *The maximum value λ of the linear function*

$$\tfrac{1}{2} \sum_{s \in T} x(\delta(s)) \qquad (65)$$

over P is equal to the maximum number μ of edge–disjoint T–paths in G.

The inequality $\lambda \geq \mu$ is obvious since an incidence vector of any family of T–paths satisfies (64) and has (65) equal to μ.

To see the equality, by Mader's edge–disjoint T–paths theorem there exist disjoint sets X_s for each $s \in T$ such that $s \in X_s$ and

$$\mu = \tfrac{1}{2}\Big(\sum_{s\in T} d_E(X_s) - \kappa\Big) \tag{66}$$

where κ is the number of components K of the graph $G' = G - \cup_{s\in T} X_s$ with $d(K)$ odd. Let $W = V(G')$ and $\mathcal{K}$ be the collection of components of G' and let F be the set of edges connecting different sets of X_s. We will call $K \in \mathcal{K}$ odd if $d_E(K)$ is odd and we call it even if $d_E(K)$ is even. Then

$$\begin{aligned}
2\mu &= \sum_{s\in T} d_E(X_s) - \kappa = 2|F| + \sum_{K\in\mathcal{K}} 2\lfloor \tfrac{1}{2} d_E(K)\rfloor \\
&= 2|F| + \sum_{\substack{K\in\mathcal{K}\\ K \text{ even}}} d_E(K) + \sum_{\substack{K\in\mathcal{K}\\ K \text{ odd}}} 2\lfloor \tfrac{1}{2} d_E(K)\rfloor \\
&\geq^{(64b)} 2|F| + \sum_{\substack{K\in\mathcal{K}\\ K \text{ even}}} x(\delta(K)) + \sum_{\substack{K\in\mathcal{K}\\ K \text{ odd}}} x(\delta(K)) \\
&= 2|F| + \sum_{K\in\mathcal{K}} x(\delta(K)) \geq 2x(F) + \sum_{K\in\mathcal{K}} x(\delta(K)) \\
&= 2x(F) + x(\delta(W)) = 2x(F) + x(\delta(\cup_{s\in T} X_s)) \\
&= \sum_{x\in T} x(\delta(X_s)) \geq^{(64c)} \sum_{s\in T} x(\delta(s)) = 2\lambda
\end{aligned} \tag{67}$$

concluding $\mu \geq \lambda$ and hence $\mu = \lambda$. This implies that μ can be determined in polynomial time. To obtain the family of paths we can use an approach similar to the one from the previous section.

This approach can be even extended to solve capacitated version of this problem: given a capacity function $c : E \to \mathbb{Z}_+$ find the maximum number of T–paths such that they together use each edge e at most $c(e)$ times.

4. Combinatorial algorithm for maximum S–path packing

The first combinatorial algorithm (not using matroids) for finding maximum packing of T–paths for the original Mader's problem was presented by Chudnovsky, Cunningham and Geelen in [**CCG04**]. They have found an algorithm for finding a maximum packing of non–zero T–paths in group–labeled graphs — a generalization of the original problem.

Later Pap came in [**Pap05a**] with an algorithm for maximum packing of non–returning T–paths in permutation–labeled graphs, further generalizing the above mentioned algorithm.

CHAPTER 5

Results

1. Polyhedral description of T-paths

One of the goals of this paper was to derive polyhedral description or linear programming formulation of the Mader's T-paths problem. That basically means finding some reasonably formulated linear programming problem that has the same optimum value as the Mader's path-packing problems. Such a thing was for instance done by Keijsper, Pendavingh, Stougie in [**KPS06**] for edge-disjoint T-paths — where they have also shown polynomial-time separability for that description and therefore polynomial-time solvability of such a problem by ellipsoid method.

Let G be a graph and $T \subseteq V(G)$. The fully vertex-disjoint T-paths packings polyhedron $P_{G,T}$ is then the convex hull of incidence vectors of all fully vertex-disjoint T-path packings in G. There is generally not much hope for finding a "nice" description for $P_{G,T}$ that would also come with a separation algorithm for the conditions. More precisely

OBSERVATION 5.1. *There is no description of $P_{G,T}$ such that $P_{G,T} = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ and all the entries of A, b are polynomial in n, together with an algorithm polynomial in n for separation from $P_{G,T}$ (unless $P = NP$).*

PROOF. If there was such a description and algorithm we show we would be able to decide whether a given graph has a Hamiltonian path in polynomial time. Suppose there is a separation algorithm. Let $G = (V, E)$ be a graph. Given two vertices $s, t \in V$ we will show that we are able to answer the query whether there is an s–t path traversing all the vertices of V. Let $T = \{s, t\}$. Then there are no two disjoint T-paths in G. Consider the linear program $\max\{\mathbf{1}^T x \mid x \in P_{G,T}\}$. Since $P_{G,T}$ is a polytope, the maximum is attained by a vertex that corresponds to a fully vertex-disjoint paths packing. We are maximizing the total number of edges and therefore the optimum is the length of the longest s–t path in G. Since we have a polynomial separation algorithm, we can compute this value in polynomial time by the ellipsoid method. Then there is a Hamiltonian path connecting s and t if and only if the optimum of our linear program is $|V| - 1$. We can easily test this for all pairs s, t and therefore get a polynomial-time algorithm for Hamiltonian path. □

Similar result can also be easily seen for internally vertex–disjoint T–paths packings and for vertex–disjoint $\mathcal{S}$–paths packings.

Also for a directed version of arc–disjoint T–paths the same result can be shown

OBSERVATION 5.2. *There is no description of an arc–disjoint T–paths packings polytope $P^{\uparrow}_{G,T}$ such that $P^{\uparrow}_{G,T} = \{x \in \mathbb{R}^n | Ax \leq b\}$ where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ and all the entries of A, b are polynomial in n, together with an algorithm polynomial in n for separation from $P_{G,T}$ (unless $P = NP$).*

PROOF. We can easily simulate the maximum fully vertex–disjoint T–paths packing problem by the maximum arc–disjoint T–paths packing problem. We split each vertex v into two vertices v^{in} and v^{out} connected by an arc $a_v = v^{\text{in}}v^{\text{out}}$, where all arcs uv from the original graph now become $u^{\text{out}}v^{\text{in}}$. Additionally we add a new vertex t' on a_t for each $t \in T$ thus making edges $t^{\text{in}}t'$ and $t't^{\text{out}}$. This reduction together with the previous observation proves the statement. □

Therefore there is just hope either for a general description of the polytopes that is not polynomially separable or for a description that is problem specific and exploits the problem structure in some way.

In the following subsections we show a description for vertex–disjoint case that was considered in Gallai's theorem.

1.1. b–factors. The first and most straightforward approach was to derive this formulation from some already known results about similar topics. In the case of maximum fully vertex–disjoint paths packing the problem has a quite simple structure that can be described through matchings or even more easily by simple b–matchings or b–factors.

We can describe it as a problem of finding a collection $\mathcal{C}$ of edges such that each edge is chosen at most once, each vertex in T has at most one of its incident edges contained in $\mathcal{C}$ and each vertex in $V - T$ has zero or two of its incident edges contained in $\mathcal{C}$. If we construct G' by taking G and adding a loop to every node in $V - T$ then the conditions concerning vertices in $V - T$ can be simplified to a condition that each vertex in $V - T$ is exactly twice amongst the endpoints of edges in $\mathcal{C}$ for G'. Let

$$b_v = \begin{cases} 1 & v \in T \\ 2 & v \in V - T \end{cases} \tag{68}$$

Then we want to find a simple b–matching in G such that the b–condition is satisfied with equality for all vertices in $V - T$ and the number of edges incident to vertices in T in the matching is maximized.

Simple b–matching polyhedron is exactly the polyhedron described by inequalities in (38). We know it is a convex hull of incidence vectors of b–matchings, hence all its vertices are b–matchings. If we consider its face described by the additional equalities

$$x(\delta(v)) = 2 \quad \text{for each } v \in V - T \tag{69}$$

then its vertices (if any) are incidence vectors of b–matching that satisfy the b–conditions for the vertices out of T. Therefore the optimal value over the resulting polyhedron of a linear function $c^T x$ such that $c := \sum_{t \in T} \lambda_{\delta(t)}$ is integral and is attained by some vertex, which means by a b–matching. After a closer look it gives exactly twice the number of endpoints of edges in the b–matching within T. That is — from previous — twice the maximum size of fully vertex–disjoint T–paths packing in G.

We will obtain the following linear program:

(70a) $0 \leq x_e \leq 1$ for $e \in E(G')$

(70b) $x(\delta_{G'}(v)) \leq 1$ for $v \in T$

(70c) $x(\delta_{G'}(v)) = 2$ for $v \in V - T$

(70d) $x(E[U]) + x(F) \leq \lfloor \frac{1}{2}(b(U) + |F|) \rfloor$ $U \subseteq V, F \subseteq \delta(U), b(U) + |F|$ odd

We can remove the conditions for loops thus obtaining again the set of conditions for G and substitute for b in the last condition. This way we obtain:

(71a) $0 \leq x_e \leq 1$ for $e \in E(G')$

(71b) $x(\delta(v)) \leq 1$ for $v \in T$

(71c) $x(\delta(v)) + 2y_v = 2$ for $v \in V - T$

(71d) $x(E[U] \cup F) + y(U - T) \leq |U - T| + \lfloor \frac{1}{2}(|U \cap T| + |F|) \rfloor$

for $U \subseteq V, F \subseteq \delta(U), |U \cap T| + |F|$ odd

This already gives an exact linear programming description of the Gallai's problem (and even its polytope) but it can be further simplified.

1.2. Pure calculations. With a slightly different and more straightforward approach it is possible to obtain simpler inequalities for the problem. Consider a problem of finding the maximum number of fully vertex–disjoint T–paths. We assume that there are no two vertices of T connected by an edge (otherwise we subdivide the edge). We will show that this problem has the same optimum value as the following linear program:

$$\frac{1}{2} \max x(\delta(T)) \tag{72}$$

subject to

$$\begin{aligned}
&(73a) && 0 \le x_e \le 1 && e \in E\\
&(73b) && 0 \le y_v \le 1 && v \in V - T\\
&(73c) && x(\delta(u)) + 2y_u = 2 && u \in V - T\\
&(73d) && x(\delta(u)) \le 1 && u \in T\\
&(73e) && x(E[U]) + y(U - T) \le |U - T| + \lfloor \tfrac{1}{2}|U \cap T| \rfloor && U \subseteq V, |U \cap T| \text{ odd}
\end{aligned}$$

Lets denote λ the optimum of this program and let μ be the optimum from Gallai's theorem. First of all we show that $\mu \le \lambda$. To do so we construct — from the fully vertex–disjoint paths family — a feasible solution of the linear program with the same value.

Let X be the set of edges used in the paths and let Y be the set of vertices in $V - T$ not used by any path. Let $x = \chi_X$ and $y = \chi_Y$. Then x, y is a feasible solution of the linear program. Conditions (73a), (73b), (73c) and (73d) are obviously valid. Let us have a look at (73e). First, we observe the following

$$\sum_{u \in U} x(\delta(u)) = 2x(E[U]) + x(\delta(U)) \tag{74}$$

Also for each set U by summing all relevant equations (73c) and (73d) we get the following inequality

$$\sum_{u \in U} x(\delta(u)) + 2y(U - T) \le 2|U - T| + |U \cap T| \tag{75}$$

From those we can derive $2x(E[U]) + 2y(U - T) \le 2x(E[U]) + x(\delta(U)) + 2y(U - T) \le^{(74)} \sum_{u \in U} x(\delta(u)) + 2y(U - T) \le^{(75)} 2|U - T| + |U \cap T|$. That shows that (73e) is implied from the other inequalities for even $|U \cap T|$, which we will find useful later.

Now consider U with $U \cap T$ odd. Since all x, y are integral, the left side is even and the right side is odd, we can sharpen the inequality to $2x(E[U]) + 2y(U - T) \le 2|U - T| + (|U \cap T| - 1)$, which is the same as (73e).

Additionally, if we have $U \subseteq T$ then (because T is a stable set) we can write

$$\sum_{u \in U} x(\delta(u)) = x(\delta(U)) \tag{76}$$

Now we are ready to prove $\mu \ge \lambda$. Let us consider the partition U from dual in Gallai's theorem and $\mathcal{K}$ the set of resulting components of $G - U$. We will denote $\mathcal{K}_0 = \{K \in \mathcal{K} \mid |K \cap T| \text{ is even}\}$, $\mathcal{K}_1 = \{K \in \mathcal{K} \mid |K \cap T| \text{ is odd}\}$. Then we can write

$$\begin{aligned}
2\mu &= 2|U| + 2\sum_{K\in\mathcal{K}} \lfloor \tfrac{1}{2}|K\cap T|\rfloor \\
&= 2|U\cap T| + 2|U-T| + \sum_{K\in\mathcal{K}_0} |K\cap T| + \sum_{K\in\mathcal{K}_1} 2\lfloor \tfrac{1}{2}|K\cap T|\rfloor \\
&\geq^{(75)} \sum_{u\in U\cap T} x(\delta(u)) + \sum_{u\in U} x(\delta(u)) + 2y(U-T) + \\
&\qquad \sum_{K\in\mathcal{K}_0}\sum_{u\in K\cap T} x(\delta(u)) + \sum_{K\in\mathcal{K}_1} 2\lfloor \tfrac{1}{2}|K\cap T|\rfloor \\
&=^{(76)} x(\delta(U\cap T)) + \sum_{K\in\mathcal{K}_0} x(\delta(K\cap T)) + \sum_{u\in U} x(\delta(u)) + \\
&\qquad 2y(U-T) + \sum_{K\in\mathcal{K}_1} 2\lfloor \tfrac{1}{2}|K\cap T|\rfloor \\
&\geq x(\delta(U\cap T)) + \sum_{K\in\mathcal{K}_0} x(\delta(K\cap T)) + x(\delta(U)) + \sum_{K\in\mathcal{K}_1} 2\lfloor \tfrac{1}{2}|K\cap T|\rfloor \\
&\geq^{(73e)} x(\delta(U\cap T)) + \sum_{K\in\mathcal{K}_0} x(\delta(K\cap T)) + x(\delta(U)) + \\
&\qquad \sum_{K\in\mathcal{K}_1} 2(x(E[K]) + y(K-T) - |K-T|) \\
&\geq x(\delta(U\cap T)) + \sum_{K\in\mathcal{K}_0} x(\delta(K\cap T)) + \sum_{K\in\mathcal{K}_1} x(\delta(K)) + \\
&\qquad \sum_{K\in\mathcal{K}_1} 2(x(E[K]) + y(K-T) - |K-T|) \\
&=^{(74)} x(\delta(U\cap T)) + \sum_{K\in\mathcal{K}_0} x(\delta(K\cap T)) + \\
&\qquad \sum_{K\in\mathcal{K}_1} \Big(\sum_{u\in K} x(\delta(u)) + 2y(K-T) - 2|K-T|\Big) \\
&=^{(73c)} x(\delta(U\cap T)) + \sum_{K\in\mathcal{K}_0} x(\delta(K\cap T)) + \sum_{K\in\mathcal{K}_1}\sum_{u\in K\cap T} x(\delta(u)) \\
&=^{(76)} x(\delta(U\cap T)) + \sum_{K\in\mathcal{K}_0} x(\delta(K\cap T)) + \sum_{K\in\mathcal{K}_1} x(\delta(K\cap T)) \\
&=^{(76)} x(\delta(T)) = 2\lambda
\end{aligned}$$

1.3. Duality approach. The same problem can be approached differently to further simplify the last inequality of the linear program. Consider the following linear program

$$\text{(77)} \qquad \tfrac{1}{2}\max x(\delta(T))$$

subject to

$$\begin{array}{lll} \text{(78a)} & x(\delta(u)) \leq 1 & u \in T \\ \text{(78b)} & x(\delta(u)) \leq 2 & u \in V - T \\ \text{(78c)} & x(\delta(R,S)) - x(\delta(R, V - (R \cup S))) \leq 2\lfloor \tfrac{1}{2}|S| \rfloor & S \subseteq T, R \subseteq V - T \\ \text{(78d)} & x \geq 0 & \end{array}$$

This can be derived from (73) by substituting for the slack variable y and adding some implied inequalities. The inequalities (78a), (78b) and (78d) are implied by (73c) and (73d). The last inequality to show is (78c). At first let us remind that (73e) is valid in (73) also without the odd condition as we have shown earlier. We will now derive (78c) from this extended (73e) that says:

$$\text{(79)} \qquad x(E[U]) + y(U - T) \leq |U - T| + \lfloor \tfrac{1}{2}|U \cap T| \rfloor \text{ for } U \subseteq V$$

we substitute from (73d) to get

$$\text{(80)} \qquad 2x(E[U]) \leq \sum_{u \in U - T} x(\delta(u)) + 2\lfloor \tfrac{1}{2}|U \cap T| \rfloor$$

now we cluster the edges and since there are no edges within T we get exactly

$$\text{(81)} \qquad 2x(E[U - T]) + 2x(\delta(U \cap T, U - T)) - \sum_{u \in U - T} x(\delta(u)) \leq 2\lfloor \tfrac{1}{2}|U \cap T| \rfloor$$

this can be rewritten from (74) to

$$\text{(82)} \qquad 2x(\delta(U \cap T, U - T)) - x(\delta(U - T)) \leq 2\lfloor \tfrac{1}{2}|U \cap T| \rfloor$$

which is, after regrouping the edges and denoting $S = U \cap T$ and $R = U - T$, exactly (78c).

Therefore the description (78) implies the description (73). It is again easy to see that if we have a maximal paths packing of size μ from Gallai's theorem it again gives us a feasible solution of (78) attaining value μ (for example because we were able to derive it for the previous linear program). Let λ be the optimal value of (77). The previous gives us $\lambda \geq \mu$. To prove $\mu \geq \lambda$ consider the dual program of (77), which is

$$\text{(83)} \qquad \tfrac{1}{2}\Big(\min \sum_{u \in T} \alpha_u + 2 \sum_{u \in V - T} \beta_u + 2 \sum_{\substack{S \subseteq T \\ R \subseteq V - T}} \sigma_{R,S} \lfloor \tfrac{1}{2}|S| \rfloor \Big)$$

subject to

(84a)

$$\underbrace{\sum_{u\in e\cap T}\alpha_u+\sum_{u\in e-T}\beta_u}_{\square}+\underbrace{\sum_{\substack{R\subseteq V-T\\ e\in\delta(R)}}\Big(\sum_{\substack{S\subseteq T\\ e\in\delta(S)}}\sigma_{R,S}-\sum_{\substack{S\subseteq T\\ e\notin\delta(S)}}\sigma_{R,S}\Big)}_{\star}\geq\begin{cases}1 & e\in\delta(T)\\ 0 & e\in E-\delta(T)\end{cases}$$

(84b)

$$\alpha,\beta,\sigma\geq 0$$

We will show that from Gallai's theorem we will find a solution of this dual attaining the value μ, which will give us the desired inequality.

Let U be the set attaining the optimal value in Gallai's theorem in dual and let $\mathcal{K}$ set of components of $G-U$. Let

$$\alpha_u:=\begin{cases}2 & u\in U\cap T\\ 0 & \text{otherwise}\end{cases}\tag{85}$$

$$\beta_u:=\begin{cases}1 & u\in U-T\\ 0 & \text{otherwise}\end{cases}\tag{86}$$

$$\sigma_{R,S}:=\begin{cases}1 & K\in\mathcal{K}, K=R\cup S, R=K-T, S=K\cap T\\ 0 & \text{otherwise}\end{cases}\tag{87}$$

In (83) this assignment attains the value $\frac{1}{2}\big(2|U\cup T|+2|U-T|+2\sum_{K\in\mathcal{K}}\lfloor\frac{1}{2}|K\cap T|\rfloor\big)$, which is exactly μ. Hence if we show this is a feasible solution we are finished. We will discuss the situation according to e.

If e connects two vertices in U then the left hand side is always at least 2 because $(\star)=0$ in this case.

If e connects U with a component $K\in\mathcal{K}$ then $(\star)\geq 0$ if $e\cap K\subseteq T$ and $(\star)\geq -1$ otherwise. For $\square$ we have $(\square)\geq 2$ if $e\cap U\subseteq T$ and $(\square)\geq 1$ otherwise. So we get that total $(\square)+(\star)\geq 0$ and if at least one end of e is in T then $(\square)+(\star)\geq 1$, so the conditions are satisfied.

The last case is that e connects two vertices in a single $K\in\mathcal{K}$. If none of them is in T then $(\star)=0$, otherwise $(\star)=1$. $(\square)$ is always zero.

So our conditions are satisfied for all possible positions of e and hence our solution is dual feasible, which gives us $\mu\geq\lambda$ and together with previous also $\mu=\lambda$.

2. Disjoint returning T–paths

Another goal of this thesis was to try to generalize the Pap's result about non–returning paths form [**Pap05b**].

The first attempt could be done on a slightly different concept — returning T–paths:

Consider a permutation–labeled graph $\mathcal{G} = (G, T, \Omega, \omega, \pi)$. A path $P = (v_0 e_1 v_1 \dots e_k v_k)$ in G is called *returning* if $\pi(P)(\omega(v_0)) = \omega(v_k)$. We can have a brief look at the problem of finding a maximum–size family of returning T–paths in $\mathcal{G}$ and show it is already NP–complete.

We will reduce from a disjoint paths problem. Given an instance of disjoint paths problem $G, \{(s_1, t_1), \dots, (s_k, t_k)\}$, we are going to construct an instance of disjoint returning T–paths problem. Let G' arise from G by adding new vertices $\{p_i\}_{i=1}^k, \{q_i\}_{i=1}^k$ and all edges $s_i p_i$ and $t_i q_i$. Let $T = \{p_i\}_{i=1}^k \cup \{q_i\}_{i=1}^k$, $\Omega = [k]$ and $\omega(x) = 1$ for all x in T. We orient all edges in G' arbitrarily. Let

$$\pi(uv) := \begin{cases} (1, i) & uv = s_i p_i \vee uv = t_i q_i \\ \text{id} & \text{otherwise} \end{cases}$$

Now let $\mathcal{G} = (G', T, \Omega, \omega, \pi)$. It is easy to see that there are disjoint paths between all given pairs of vertices in G if and only if there exist k disjoint returning T–paths in $\mathcal{G}$. This gives us NP–hardness of the problem. Since the paths themselves are polynomial in the size of input, it immediately gives NP–completeness of the problem.

3. Other non–zero and non–returning T–paths packings

Proofs of Mader's min–max relation for $\mathcal{S}$–paths and of min–max relation for non–returning T–paths in permutation–labeled graphs have one element in common. They are showing and exploiting some kind of exchange properties of those path packings — or in the case of non–returning paths using only weaker property that every path packing can be extended to a larger one (if there is some), while preserving all the path endpoints in T. The proof of Mader's min–max relation shows the exchange property for path endpoints that forms a Mader matroid. Pap's proof of min–max relation for non–returning T–paths also shows and implies the following:

THEOREM 5.3 (Exchange property for non–returning paths). *Let* $Y = \text{ends}_V(\mathcal{Q})$ *for some family* $\mathcal{Q}$ *of fully vertex–disjoint non–returning* T*–paths. Let* $Z = \text{ends}_V(\mathcal{R})$ *for a fully vertex–disjoint* T*–paths family* $\mathcal{R}$ *such that* $|Y| < |Z|$*. Then there exist* X *such that* $X = Y \cup \{s, t\}$ *for some* $s, t \in Z - Y$ *and* $X = \text{ends}_V(\mathcal{P})$ *for some family* $\mathcal{P}$ *of fully vertex–disjoint non–returning* T*–paths in* $\mathcal{G}$*.*

Similarly as in [**Pap05b**], the proof follows:

PROOF. We prove by induction on $|V|$. Let $\mathcal{Q}$ be a non–returning family with $Y = \text{ends}_V(\mathcal{Q})$ and $\mathcal{R}$ a non–returning family with $Z = \text{ends}_V(\mathcal{R})$ such that $|Y| < |Z|$. Without loss of generality we can assume $|\mathcal{R}| = |\mathcal{Q}| + 1$ and therefore $|Z| = |Y| + 2$.

If $Y \subseteq Z$ we are finished since we can take $\mathcal{P} := \mathcal{R}$ and $X := Z$. Otherwise there is a node $r \in Y - Z$.

If r is covered in $\mathcal{Q}$ by a path of length one rr' then $Y' = Y - r - r'$ satisfies $Y' = \text{ends}_V(\mathcal{P}')$ for a non-returning family $\mathcal{Q}' = \mathcal{Q} - rr'$ in $\mathcal{G}' = (G - r - r', T - r - r')$. Let $\mathcal{R}'$ arise from $\mathcal{R}$ by deleting the path starting in r' (if there is any) and let $Z' := \text{ends}_V(\mathcal{R}')$. $\mathcal{R}'$ is a non-returning family in $\mathcal{G}'$ and $|Y'| < |Z'|$. Since $|V(G')| < |V(G)|$ we can use induction — so there are $s, t \in Z' \subseteq Z$ such that there is a non-returning family $\mathcal{P}'$ in $\mathcal{G}'$ with $X' := Y' + \{s, t\} = \text{ends}_V(\mathcal{P}')$. Now $X := Y + \{s, t\}$ satisfies $X = \text{ends}_V(\mathcal{P})$ for $\mathcal{P} := \mathcal{P}' \cup rr'$ — obtained from $\mathcal{P}'$ by adding back the path rr'. So $\mathcal{P}$ is the desired family.

Otherwise r is covered in $\mathcal{Q}$ by a path of length more than 1 starting by an edge $rq \in E, q \in V - T$. Define $\omega' : T - r + q \to \Omega$ by ω on $T - r$ and $\omega'(q) := \pi(rq, r)(\omega(r))$. Then $Y' := Y - r + q$ is $Y' = \text{ends}_V(\mathcal{Q}')$ where $\mathcal{Q}'$ arises from $\mathcal{Q}$ by shortening the path ending in r just to q. $\mathcal{Q}'$ is a non-retuning family in $\mathcal{G}' = (G - r, T - r + q, \omega', \pi)$. We claim that there is a bigger non-returning family in $\mathcal{G}'$. If none of the paths from $\mathcal{R}$ traverses q then $\mathcal{R}$ is that family. After applying induction we get a non-returning family $\mathcal{P}'$ covering X' such that $X' = Y' + \{s, t\}$ for some $s, t \in Z - Y'$. Then $\mathcal{P}$ obtained from $\mathcal{P}'$ by extending the path ending in q to r is a non-returning family in $\mathcal{G}$ covering $X := Y + \{s, t\}$ where $s, t \in Z - Y' = Z - Y' - r \subseteq Z - Y$. Otherwise let $R \in \mathcal{R}$ joining q_1 to q_2 traverse r. Then one of the sections q_1–q and q–q_2 must be non-returning (otherwise the whole path is returning). Without loss of generality it is q_1–q. Let R' be this path. Let $Z' := Z - q_2 + q$ and $\mathcal{R}' := \mathcal{R} - R + R'$. Then $\mathcal{R}'$ is a non-returning family in $\mathcal{G}'$ and $Z' = \text{ends}_V(\mathcal{R}')$. So $\mathcal{R}'$ gives a bigger non-returning family. From the induction we get a family $\mathcal{P}'$ covering $X' := Y' + \{s, t\}$ for some $s, t \in Z' - Y'$. Then consider $X := Y + s + t$ and $\mathcal{P}$ arise from $\mathcal{P}'$ by extending the path ending in q to r. Then $\mathcal{P}$ is a non-returning family in $\mathcal{G}$ and $s, t \in Z' - Y' = Z' - Y' - r = (Z - q_2 + q) - (Y - r + q) \subseteq Z - Y$. That completes the proof. □

In the Pap's min–max result only weaker result is shown and for the rest of the proof needed.

COROLLARY 5.4 (Extendability of non-returning T-paths packings). *Let $Y = \text{ends}_V(\mathcal{Q})$ for some family $\mathcal{Q}$ of fully vertex-disjoint non-returning T-paths, where $\mathcal{Q}$ is not maximal-size such packing. Then there is $\mathcal{Q}'$ with $Y' := \text{ends}_V(\mathcal{Q}')$ such that $Y' = Y + s + t$ for some $s, t \in T$.*

If we think about proving some min–max relations like this one for more general non-returning internally vertex-disjoint T-paths, non-returning disjoint $\mathcal{S}$-paths or non-returning edge-disjoint T-paths, we would like to have a similar property in order to be able to apply an analogous approach.

Unfortunately this is not possible since such a property does not hold for either of mentioned packings. The following statements are internally vertex-disjoint and edge-disjoint analogues to the above mentioned extendability:

Desired property (Extendability for non-returning (non-zero) internally-disjoint paths). Let $Y = \mathrm{ends}_{\mathrm{E}}(\mathcal{Q})$ for some family $\mathcal{Q}$ of internally vertex-disjoint non-returning (non-zero) T-paths. Let $\mathcal{R}$ be an internally vertex-disjoint non-returning (non-zero) T-paths family such that $|\mathcal{R}| > |\mathcal{Q}|$. Then there exists a family $\mathcal{P}$ of internally vertex-disjoint non-returning (non-zero) T-paths in $\mathcal{G}$ such that $\mathrm{ends}_{\mathrm{E}}(\mathcal{P}) = Y + s + t$ for some $s, t \notin Y$ edges with one endpoint in T.

and

Desired property (Extendability for non-returning (non-zero) edge-disjoint paths). Let $Y = \mathrm{ends}_{\mathrm{E}}(\mathcal{Q})$ for some family $\mathcal{Q}$ of edge-disjoint non-returning (non-zero) T-paths. Let $\mathcal{R}$ be an edge-disjoint non-returning (non-zero) T-paths family such that $|\mathcal{R}| > |\mathcal{Q}|$. Then there exists a family $\mathcal{P}$ of edge-disjoint non-returning (non-zero) T-paths in $\mathcal{G}$ such that $\mathrm{ends}_{\mathrm{E}}(\mathcal{P}) = Y + s + t$ for some $s, t \notin Y$ edges with one endpoint in T.

Unfortunately, none of these properties hold even for the non-zero paths over a binary group $\mathbb{Z}_2$ and we give a common counterexample for them which is shown in Figure 1.

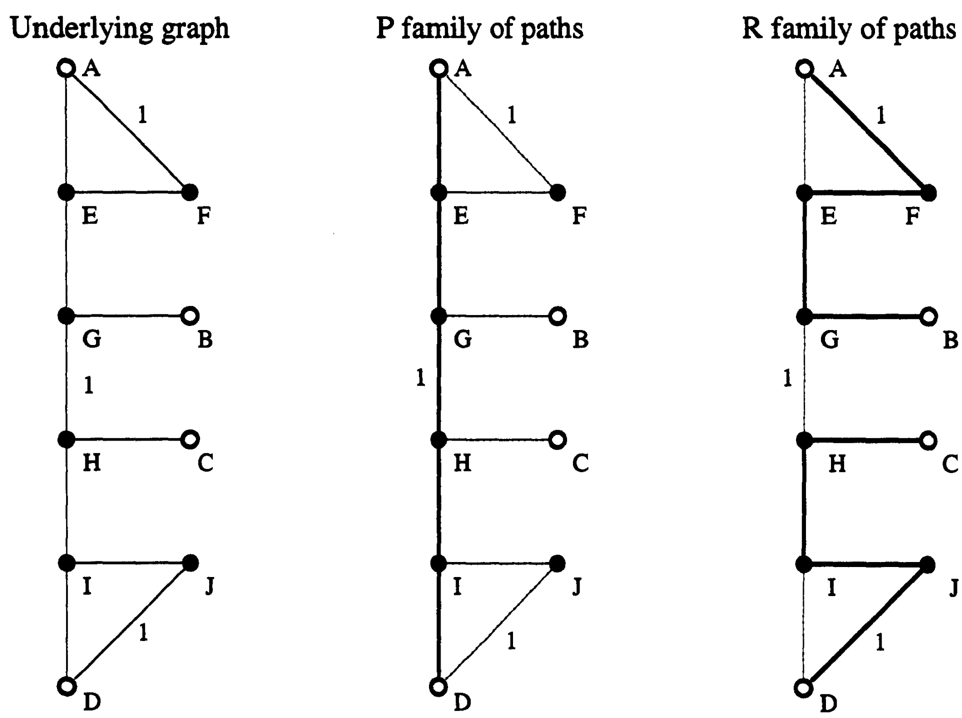

Figure 1. Edge-disjoint and internally vertex-disjoint counterexample

V consists of A, B, C, D, E, F, G, H, I and J. The set of terminals is $T = \{A, B, C, D\}$. Group Γ is $\mathbb{Z}_2$. Since we are in a binary group we do not need to take into account orientation of the edges since the parity is the same in both ways. All edges e have $\gamma(e) := 0_\Gamma$ with the exception of $e \in \{AF, GH, ID\}$, where it is $\gamma(e) := 1_\Gamma$. From now on we will call those edges e with $\gamma(e) = 1_\Gamma$ *parity edges.* Let family $\mathcal{P}$ consist of a path $P_1 = AEGHID$ and family $\mathcal{R}$ of paths $R_1 = AFEGB$ and $R_2 = DJIHC$.

At first we will observe that the only path that can start with an edge AE is exactly P_1. It has to traverse odd number of parity edges and thus at least one of them. Therefore this path has to traverse the edge GH. Now there is only one way to reach another vertex of T without passing even number of parity edges in total and that is exactly P_1. Now it is easy to see that this path does not allow any other path connecting vertices in T being either internally vertex–disjoint or edge–disjoint to it. So the conclusion is that $\text{ends}_{\text{E}}(\mathcal{P})$ cannot even be enlarged in any way although it is not maximal (as we can see from $\mathcal{R}$ which consists of two disjoint non–zero paths).

This counterexample already gives an idea how the counterexample for the Mader's disjoint $\mathcal{S}$–paths version will look like (note that it is sufficient to split end vertices into the sets of $\mathcal{S}$). We show it in Figure 2.

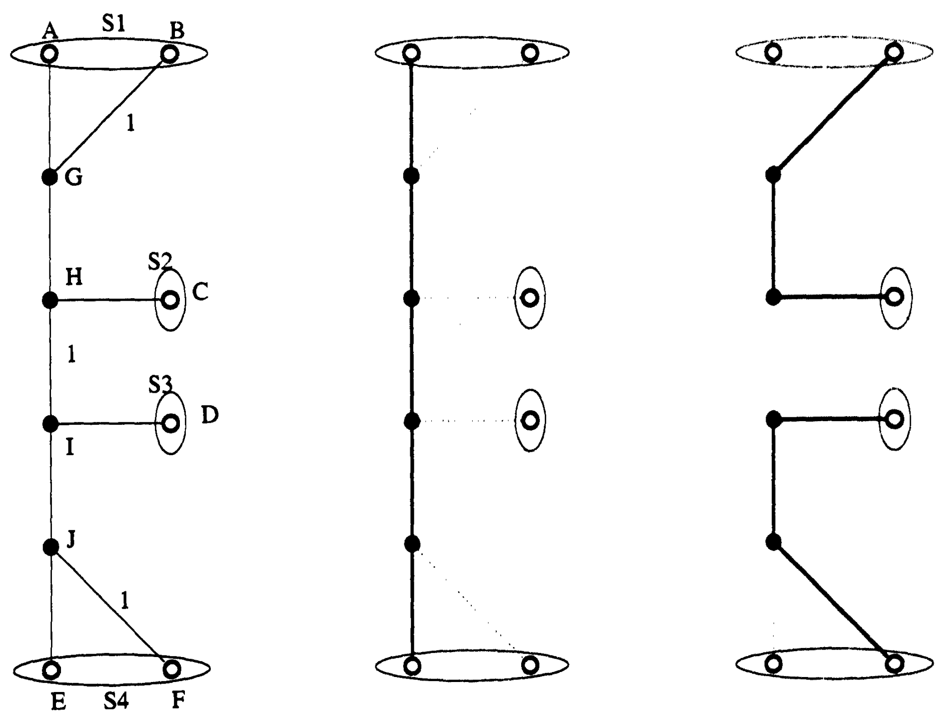

FIGURE 2. Mader counterexample

The vertex set consists of vertices A, B, C, D, E, F, G, H, I and J. The sets of terminals are $\mathcal{S} = \{\{A, B\}, \{C\}, \{D\}, \{E, F\}\}$. The group Γ is again $\mathbb{Z}_2$. All edges e have $\gamma(e) = 0_\Gamma$ except for $e \in \{BG, HI, JF\}$ having $\gamma(e) = 1_\Gamma$.

The path family $\mathcal{P}$ consists of one path $P_1 = AGHIJE$ and the family $\mathcal{R}$ of two disjoint paths $R_1 = BGHC$ and $R_2 = FJID$. We will show that P_1 and $AGHID$ are the only non–zero paths that can start (end) by AG. It follows from the fact that it has to traverse one or three parity edges. Since it cannot connect A with B (they are in the same set of $\mathcal{S}$) it must traverse IJ. Only ways to continue are either finish in D or continue as P_1. Similarly, the only paths starting (ending) with JE are P_1 and $CHIJE$. If we have a closer look at those three paths $AGHID$, $CHIJE$ and $AGHIJE$ we see that neither of them allows any other $\mathcal{S}$–path to be disjoint with it. That gives us that $\text{ends}_{\text{E}}(\mathcal{P})$ cannot be extended in any way even though there exists a larger set covered by the family $\mathcal{R}$.

One could also ask a question whether a weaker property would not be sufficient. For example requiring only increasing the number of edges at each vertex (and not extending the set of edges) would suffice to follow a part of Pap's proof. For internally disjoint T–paths we might only require

DESIRED PROPERTY (Weak extendability for non–returning (non–zero) internally vertex–disjoint paths). Let $Y = \text{ends}_{\text{E}}(\mathcal{Q})$ for some family $\mathcal{Q}$ of internally vertex–disjoint non–returning (non–zero) T–paths. Let $\mathcal{R}$ be an internally vertex–disjoint non–returning (non–zero) T–paths family such that $|\mathcal{R}| > |\mathcal{Q}|$. Then there exists a family $\mathcal{P}$ of internally vertex–disjoint non–returning (non–zero) T–paths in $\mathcal{G}$ such that for each $t \in T$ we have $|\{e \in \text{ends}_{\text{E}}(\mathcal{P}) \mid t \in e\}| \leq |\{e \in \text{ends}_{\text{E}}(\mathcal{Q}) \mid t \in e\}|$.

Similarly for edge–disjoint T–paths

DESIRED PROPERTY (Weak extendability for non–returning (non–zero) edge–disjoint paths). Let $Y = \text{ends}_{\text{E}}(\mathcal{Q})$ for some family $\mathcal{Q}$ of edge–disjoint non–returning (non–zero) T–paths. Let $\mathcal{R}$ be an edge–disjoint non–returning (non–zero) T–paths family such that $|\mathcal{R}| > |\mathcal{Q}|$. Then there exists a family $\mathcal{P}$ of edge–disjoint non–returning (non–zero) T–paths in $\mathcal{G}$ such that for each $t \in T$ we have $|\{e \in \text{ends}_{\text{E}}(\mathcal{P}) \mid t \in e\}| \leq |\{e \in \text{ends}_{\text{E}}(\mathcal{Q}) \mid t \in e\}|$.

Unfortunatelly even these properties do not hold. We present the following common counterexample (note that this works as a counterexample also to the ordinary extendability but we rather prefer to show both the counterexamples). Figure 3 shows the underlying graph and Figure 4 show the path families $\mathcal{Q}$ and $\mathcal{R}$.

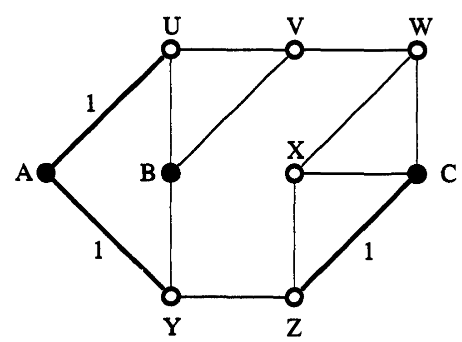

FIGURE 3. Weak extendability counterexample - underlying graph

We again consider binary group $\mathbb{Z}_2$, set $T := \{A, B, C\}$ and all edges have label 0 except for AU, AY and AZ, which have label 1.

Family $\mathcal{P}$ consists of non–zero T–paths $P_1 = AUVWC$ and $P_2 = AYZXC$, so A and C have degree 2 and B has degree 0. Now $\mathcal{R}$ consists of three non–zero T–paths $R_1 = AUB$, $R_2 = AYB$ and $R_3 = BVWXZC$. To satisfy the weak extendability, we have to find three non–zero T–paths such that at least two of

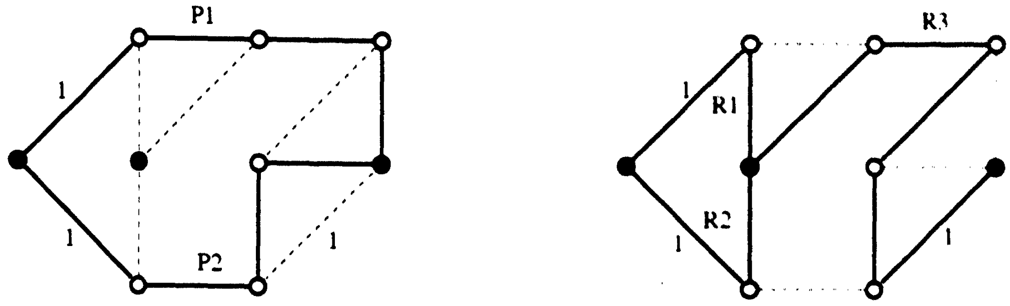

FIGURE 4. Weak extendability counterexample - the paths ($\mathcal{P}$ and $\mathcal{R}$)

them end in A and at least two in BC. Since they need to be three non-zero paths and we only have three non–zero edges, each of the paths has to use one of these edges. To satisfy the degree conditions, at least one of the paths has to connect A with C. This path can either go through A, Y, Z and X, thus preventing the edge CZ from being in a T–path or through A, U, V, W and further to C or through X to C. Then the path starting with CZ cannot use the edge VW, therefore it has to go through ZY (as VW, YZ form a cut separating A, B from C). Now from Y it has to continue to B and therefore the path starting by AY cannot be formed. This proves that the weak extendability neither holds for edge–disjoint case, nor for internally vertex–disjoint.

A counterexample for a similar property for $\mathcal{S}$–paths can be produced in a similar way as it was done with the extendability property.

The question of finding a maximum non–zero internally vertex–disjoint T–paths packing also appears to be more difficult because it covers another problem called 2 disjoint paths problem, which is the following:

Given a graph $G = (V, E)$ and two pairs of vertices (s_1, t_1) and (s_2, t_2), decide whether there exist disjoint paths P_1, P_2, such that P_i connects s_i and t_i.

A characterization when these two paths exist, together with a polynomial algorithm for deciding this problem are known due to Seymour [**Sey80**], Thomassen [**Tho80**] and Shiloach [**Shi80**]. The characterization is:

THEOREM 5.5. *Let $G = (V, E)$ be a graph and let s_1, t_1, s_2, t_2. Then G has disjoint paths P_1 and P_2, where P_i connects s_i and t_i $(i = 1, 2)$, if and only if there is no subset U of V such that:*

(1) $s_1, t_1, s_2, t_2 \in U$,
(2) $|N(K)| \leq 3$ *for each component K of $G - U$*
(3) *the graph H obtained from $G[U]$ by adding, for each component K of $G - U$ and each distinct $u, v \in N(K)$, an edge connecting u and v, is planar, with s_1, t_1, s_2, t_2 in this order cyclically on the outer boundary of some drawing of H.*

First of all, lets observe that 2–paths problem can be formulated by the non–zero T–paths packing problem. Consider $H = (V, E)$ and $s_1, t_1, s_2, t_2 \in V$ an instance of the 2 disjoint paths problem. We want to find a group–labeled graph

$\mathcal{G} = (G, T, \Gamma, \gamma)$ such that there are two internally disjoint non–zero T–paths in G if and only if the two disjoint paths in H, s_1, t_1, s_2, t_2 exist. First of all we will choose $\Gamma = \mathbb{Z}_2$ a binary group. Therefore we will not need to bother about the orientation of edges in G. Now let

(1) $V(G) := V(H) \cup \{s, t\}$ for some $s, t \notin V(H)$,
(2) $E(G) := E(H) \cup \{\{s, s_1\}, \{s, s_2\}, \{t_1, t\}, \{t_2, t\}\}$,
(3) $T := \{s, t\}$

and define

$$\gamma(e) := \begin{cases} e \in E(H) & 0 \\ e = \{s, s_1\} & 0 \\ e = \{s, s_2\} & 1 \\ e = \{t_1, t\} & 1 \\ e = \{t_2, t\} & 0 \end{cases} \tag{88}$$

Under such setting we have exactly two non–zero edges. Therefore every non–zero path uses exactly one of them. Furthermore, we have exactly two vertices in T, which gives us together with the previous that non–zero T–paths are exactly the paths starting with edge ss_1 and ending with t_1t or starting with ss_2 and ending with t_2t. Then two such paths P_1, P_2 are disjoint if and only if (wlog) $P_1 := ss_1P_1't_1t$ and $P_2 := ss_2P_2't_2t$ and the subpaths P_1' and P_2' are disjoint.

As a consequence of the previous, our min–max characterization of the non–zero internally vertex–disjoint T–paths packing would have to involve a characterization of solvability this special problem of 2 disjoint paths, which looks hopeless in the sense of expressing the planarity–related condition or something equivalent as a 'value' of the min part.

CHAPTER 6

Open questions

There are some interesting questions that might be worth of further research. Our whole research was motivated by the following problem

PROBLEM 1. Finding a linear programming formulation to a maximum number of edge–disjoint non-zero T–paths in a permutation–labeled graph.

To get an intuition about it (and also a powerful tool for proof), one would like to have a min–max relation for this quantity ([**KPS06**] used it for proving properties of their linear program for edge–disjoint T–paths as well as we did for our disjoint T–paths).

It might seem quite natural to try proving such a min–max relation for the maximum size of a non–returning edge–disjoint T–paths packing from a min–max relation for the maximum size of a non–returning internally vertex–disjoint T–paths packing, as it was done in the case of T–paths in ordinary (not permutation–labeled) graphs.

Unfortunatelly, as we have stated above, we disbelieve, that such a min–max relation can be (easily) found but on the other hand, one could try to find the min–max relation for the edge–disjoint case in a different way.

PROBLEM 2. Finding a min–max relation for the maximum size of non–returning edge–disjoint T–paths packing.

From the other end, there arise also some questions to generalize our linear programming formulation result. One could try to extend the linear programming formulation for the cardinality of maximum T–paths packing to a polynomially separable formulation:

PROBLEM 3. Finding a linear programming formulation $L(G,T)$ for the maximum number of disjoint T–paths in G, such that separation over the polyhedron (polytope) of $L(G,T)$ can be done in polynomial time.

And another possible direction might be attemping to extend our characterization to the case of internally disjoint T–paths packing, i.e.

PROBLEM 4. Finding a linear programming formulation $L(G,T)$ for the maximum number of internally vertex–disjoint T–paths in T.

A good start would be to take our program and change it in a way that it would be feasible for all (reasonable) internally–disjoint T–paths packings. This requires removing the conditions implying that we can use at most one edge at vertices in T and replacing it by conditions that would forbid paths starting and ending in the same vertex in T. Inspiration for such a condition can be found in the article [**KPS06**], where they are dealing with edge–disjoint T–paths. We have tried this approach and it seems not to lead to any linear programming formulation for this problem.

Bibliography

[Ber58] C. Berge. Sur le couplage maximum d'un graphe. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences [Paris]*, 247:258–259, 1958.

[Bir46] G. Birkhoff. Tres observaciones sobre el algebra lineal. *Revista Facultad de Ciencias Exactas, Puras y Aplicadas Universidad Nacional de Tucuman, Serie A (Matematicas y Fisicia Teorica)*, 5:147–151, 1946.

[CCG04] M. Chudnovsky, J. Cunningham, and J. Geelen. An algorithm for packing non-zero A-paths in group-labeled graphs, 2004.

[CGG+04] M. Chudnovsky, J. Geelen, B. Gerards, L. Goddyn, M. Lohman, and P. Seymour. Packing non-zero A-paths in group-labeled graphs, 2004.

[Edm65a] J. Edmonds. Maximum matching and a polyhedron with $0, 1$-vertices. *Journal of Research of the National Bureau of Standards Section B*, 69:125–130, 1965.

[Edm65b] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[EJ73] J. Edmonds and E. L. Johnson. Matching, Euler tours and the Chinese postman. *Mathematical Programming*, 5:88–124, 1973.

[Gal61] T. Gallai. Maximum-minimum Sätze und verallgemeinerte Faktoren von Graphen. *Acta Mathematica Academiae Scientiarum Hungaricae*, 12:131–173, 1961.

[GLS81] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[Kha79] L. G. Khachiyan. Polinomialnyĭ algoritm v lineĭnom programmirovanii. *Doklady Akademii Nauk SSSR*, 244(5):1093–1096, 1979.

[KPS06] J. C. M. Keijsper, R. A. Pendavingh, and L. Stougie. A linear programming formulation of Mader's edge-disjoint paths problem. *Journal of Combinatorial Theory. Series B*, 96(1):159–163, 2006.

[Lov80] L. Lovász. Matroid matching and some applications. *Journal of Combinatorial Theory. Series B*, 28(2):208–236, 1980.

[Lov81] L. Lovász. The matroid matching problem. In *Algebraic methods in graph theory, Vol. II (Szeged, 1978)*, volume 25 of *Colloquia Mathematica Societatis János Bolyai*, pages 495–517. North-Holland, Amsterdam, 1981.

[Mad78a] W. Mader. Über die Maximalzahl kantendisjunkter A-Wege. *Archiv der Mathematik (Basel)*, 30:325–336, 1978.

[Mad78b] W. Mader. Über die Maximalzahl kreuzungsfreier H-Wege. *Archiv der Mathematik (Basel)*, 31(4):387–402, 1978.

[Oxl92] J. G. Oxley. *Matroid theory*. Oxford Science Publications. The Clarendon Press Oxford University Press, New York, 1992.

[Pap94] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.

[Pap05a] G. Pap. Alternating paths revisited IV: packing and 2-packing of A-paths. Technical Report EGRES TR-2005-15, Eötvös Loránd Tudományegyetem (ELTE), 2005.

[Pap05b] G. Pap. Packing non-returning A-paths. Technical Report EGRES TR-2005-12, Eötvös Loránd Tudományegyetem (ELTE), 2005.

[Pap06] G. Pap. Mader matroids are gammoids. Technical Report TR000, Eötvös Loránd Tudományegyetem (ELTE), 2006.

[Pen] R. Pendavingh. Advanced linear programming. Lecture notes.

[Sch86] A. Schrijver. *Theory of linear and integer programming.* Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons Ltd., Chichester, 1986. , A Wiley-Interscience Publication.

[Sch01] A. Schrijver. A short proof of Mader's S-paths theorem. *Journal of Combinatorial Theory. Series B*, 82(2):319–321, 2001.

[Sch03a] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. A*, volume 24 of *Algorithms and Combinatorics.* Springer-Verlag, Berlin, 2003. Paths, flows, matchings, Chapters 1–38.

[Sch03b] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. B*, volume 24 of *Algorithms and Combinatorics.* Springer-Verlag, Berlin, 2003. Matroids, trees, stable sets, Chapters 39–69.

[Sch03c] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. C*, volume 24 of *Algorithms and Combinatorics.* Springer-Verlag, Berlin, 2003. Disjoint paths, hypergraphs, Chapters 70–83.

[Sey80] P. D. Seymour. Disjoint paths in graphs. *Discrete Mathematics*, 29(3):293–309, 1980.

[Shi80] Y. Shiloach. A polynomial solution to the undirected two paths problem. *Journal of the Association for Computing Machinery*, 27(3):445–456, 1980.

[SS04] A. Sebő and L. Szegő. The path-packing structure of graphs. In *Integer programming and combinatorial optimization*, volume 3064 of *Lecture Notes in Computer Science*, pages 256–270. Springer, Berlin, 2004.

[Tho80] C. Thomassen. 2-linked graphs. *European Journal of Combinatorics*, 1(4):371–378, 1980.

[Tut47] W. T. Tutte. The factorization of linear graphs. *The Journal of the London Mathematical Society*, 22:107–111, 1947.

[Tut54] W. T. Tutte. A short proof of the factor theorem for finite graphs. *Canadian Journal of Mathematics*, 6:347–352, 1954.