Polyhedral Aspects of Combinatorial Designs

by

Lucia Moura

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Polyhedral Aspects of Combinatorial Designs

Lucia Moura

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

1999

A t–(v,k,λ) *design* is a collection of k-subsets (called *blocks*) of a v-set such that every t-subset of the v-set is contained in *exactly* λ blocks; t–(v,k,λ) *packing designs* and *covering designs* are defined by replacing the word *exactly* by *at most* and *at least*, respectively. The polyhedron associated with a design is the convex hull of the incidence vectors $x \in \mathbb{R}^{\binom{v}{k}}$ of all designs of that kind. Given $s \geq 2$ and $v \geq k \geq t \geq 1$, a family $\mathcal{A}$ of k-subsets of $[1,v]$ is said to be *s-wise t-intersecting*, if any s members $A_1, \ldots, A_s$ of $\mathcal{A}$ are such that $|A_1 \cap \ldots \cap A_s| \geq t$; Let us denote by $I^s(v,k,t)$ the set of all such families.

The thesis is divided into three parts. In the first part, a theoretical investigation of design polytopes is undertaken. Let $T_{t,v,k,\lambda}$ and $P_{t,v,k,\lambda}$ denote the polyhedra associated with t–(v,k,λ) designs and packings, respectively. Maximal clique facets for $P_{t,v,k,\lambda}$ are characterized as maximal $(\lambda+1)$-wise t-intersecting families of k-sets of a v-set. Sub-packing inequalities for $P_{t,v,k,\lambda}$ are derived and conditions under which these inequalities define facets are given. We provide m-sparse facets for the m-sparse 2-$(v,k,1)$ packing polytope. We also show that if the difference of two designs is a (t,k)-pod, a null design of minimum support, the designs are adjacent as vertices of the $T_{t,v,k,1}$ polytope.

In the second part, we investigate the following problem: "Given $s \geq 2$, $v > k > t$, classify all maximal (with respect to set inclusion) families in $I^s(v,k,t)$". This is a wide open problem in extremal set theory. We solve the classification problem for the first nontrivial case, namely $s = 2$, $k = t+2$ and arbitrary v. We also prove that the classification problem does not depend on v. In order to derive these results, properties relating kernels and generating sets of maximal families are proven, and a construction of maximal families in $I^s(v+1, k+1, t+1)$ based on the ones in $I^s(v,k,t)$ is given.

The third part of the thesis focuses on polyhedral algorithms for constructing t-$(v,k,1)$ designs and packings, and m-sparse triple systems with $m = 4, 5$. A branch-and-cut algorithm is proposed and an implementation is described. Separation algorithms for cliques and m-sparse inequalities are presented. A partial isomorph rejection scheme is employed to avoid processing isomorphic subproblems in the branch-and-cut tree. The effects of various parameters on performance are analyzed through experiments. Our method is competitive with many other techniques for generating designs including backtracking and randomized search. Our algorithm produces new maximal cyclic t-$(v,k,1)$ packings for $t = 2, 3, 4, 5$, $k = t+1, t+2$ and small v.

Acknowledgements

First of all, I would like to thank my supervisor Rudi Mathon for his help, support and enthusiasm. It has been a pleasure to work with him.

I would also like to thank my other committee members: Allan Borodin, Mike Carter, Charlie Colbourn, Derek Corneil, Eric Mendelsohn and Mike Molloy, for several suggestions that improved this thesis. In particular, I would like to thank Charlie Colbourn for being my external examiner and for his careful reading, feedback and interesting suggestions for future work.

Daniel Panario and Brett Stevens proofread drafts of the thesis. Thanks to Ralf Borndörfer and Terry Griggs who were both very kind pointing out useful references and with whom I discussed topics related to the thesis.

Friends with whom I shared this combinatorial journey and which are also great human beings are Luis Dissett, Ali Mahmoodi, and Brett Stevens. Another friend and source of inspiration was my master's supervisor Carlos Humes who was responsible for getting me addicted to this business of doing research.

My experience at the Department of Computer Science has been very nice in many respects. First, I made many good friends (which I won't name all)... Then, I met several people that go beyond their job description, and are able to be incredibly active and human at the same time; I am thinking of Martha Hendriks, Jim Clarke and Derek Corneil. Other dear human beings I have met are Rudi Mathon and Eric Mendelsohn.

I would also like to thank my parents Wanda and Álvaro Moura, my brother Pedro and sisters Marcia and Rejane. It is hard to know who I would be now without having had you around, so I won't try to figure out specific things to mention.

Finally, I would like to acknowledge my co-authors in this joint enterprise of going overseas: Natan and Daniel. During the past years I had the privilege of having this happy little boy bringing magic and fairy tales to my life. Natan: sometimes I think you are too wise for an eight year old... My other great source of happiness and encouragement was you, Daniel. It was very lucky of me having you there for support, advice, enthusiasm, companionship and even technical discussions. Let us see where life will take us next.

Contents

Chapter 1

Introduction

1.1 Motivation and thesis overview

This thesis is devoted to a polyhedral study of combinatorial designs. In particular, we investigate properties of polytopes associated with design problems, with an aim of using this knowledge to design efficient algorithms for constructing designs.

Combinatorial design theory comprises the study of set systems with some "balance" properties. Early research in this area includes the "36 officers problem" introduced by Euler in 1782, and the works of Kirkman, Steiner and Cayley on triple systems and Room squares, in the 19th century. In this century, statistical applications led to a renewed interest in these objects (see [98]). More recently, applications in many other areas have been explored (see [24, Part V]). Applications to computer science are found in such diverse problems as file organization, distributed consensus, sorting in rounds, interconnection networks and software testing [23] [24] [27]. Other important areas of application are coding theory and cryptography.

Design theorists are concerned with existence, enumeration and analysis of a wide variety of combinatorial designs. Computational techniques play an important role in many aspects of combinatorial design theory. Infinite families of designs are usually produced by using recursive constructions that require the knowledge of a few "starter" designs. Computers have been essential in the search for these starter designs [102]. Another important role of computational methods is in the enumeration of designs by exhaustive techniques, producing a list of all designs of certain types. This is particularly important for investigating design properties and formulating conjectures [71].

Polyhedral theory has been applied to algorithms for solving hard combinatorial optimization problems. The traveling salesman problem is an example of a successful application of polyhedral algorithms [83] [84]. Little research has been done, however, in the application of these techniques to combinatorial designs. To the best of our knowledge, the first publication in this direction was the master's thesis by Wengrzik [103], in which the problem of constructing balanced incomplete block designs is approached with polyhedral theory and algorithms. Independently, we started an investigation of similar problems. In [76], we proposed in general terms, a polyhedral approach to design problems, using a different formulation and dealing with a more general class of designs, namely t-designs, packings and coverings. In this thesis, we proceed with the polyhedral study of these problems, and develop and implement an algorithm for finding designs using a branch-and-cut framework. Along the way, other interesting combinatorial problems emerged. The most evident instance was the classification of s-wise t-intersecting families of sets, which arose in the study of a class of facets for the packing design polytopes. The study of this problem led to results relevant to extremal set theory. Some of these theoretical results had algorithmic implications, and in particular, led to an efficient algorithm for facet separation.

In order to describe our results at a more concrete level, we need some definitions. First, we define the combinatorial design problems we are interested in, then describe a polyhedral reformulation of these problems, and finally outline the main contributions of this thesis.

Let us recall the definitions of t-designs, packings and coverings. A t–(v,k,λ) *design* is a pair $(V,\mathcal{B})$ where V is a v-set and $\mathcal{B}$ is a collection of k-subsets of V called *blocks* such that every t-subset of V is contained in exactly λ blocks of $\mathcal{B}$. A t–(v,k,λ) *packing design* and a t–(v,k,λ) *covering design* are defined by replacing the condition "in exactly λ blocks" in the above definition by "in at most λ blocks" and "in at least λ blocks", respectively. The *packing number*, denoted by $D_\lambda(v,k,t)$, is the *maximum* number of blocks in a t–(v,k,λ) packing design. The *covering number*, denoted by $C_\lambda(v,k,t)$, is the *minimum* number of blocks in a t–(v, k, λ) covering design. It is well known that a t–(v, k, λ) design is both a maximum packing and a minimum covering design.

Central questions in combinatorial design theory are concerned with the existence of t-designs and the determination of maximum packing designs and minimum covering designs. The same questions are relevant when additional properties are required of these designs.

We concentrate on designs without repeated blocks, known as *simple* designs. Such a design can be represented by an *incidence vector*, that is a 0-1 vector $x \in \mathbb{R}^{\binom{v}{k}}$ indexed by the k-subsets of a v-set and such that $x_S = 1$ if and only if S is a block of the design. The polyhedron associated with a design is defined as the convex hull of the incidence vectors of all designs of that kind. Let us denote by $T_{t,v,k,\lambda}$, $P_{t,v,k,\lambda}$ and $C_{t,v,k,\lambda}$ the polyhedra associated with the t–(v, k, λ) designs, packing designs and covering designs, respectively. Let us introduce a new notation for the packing and covering numbers for simple packings and coverings as $\overline{D}_\lambda(v, k, t)$ and $\overline{C}_\lambda(v, k, t)$, respectively.

The main questions regarding these designs get translated into the polyhedral world as follows. A t–(v, k, λ) design exists if and only if the polyhedron $T_{t,v,k,\lambda}$ is nonempty; the incidence vectors of the t–(v, k, λ) designs are precisely the vertices of this polyhedron. The packing number is rewritten as $\overline{D}_\lambda(v, k, t) = \max\{\mathbf{1}^T x : x \in P_{t,v,k,\lambda}\}$ and the corresponding maximum solutions x that are vertices of $P_{t,v,k,\lambda}$ correspond to the maximum packings. Analogously, the covering number is rewritten as $\overline{C}_\lambda(v, k, t) = \min\{\mathbf{1}^T x : x \in C_{t,v,k,\lambda}\}$ and the corresponding minimum solutions x that are vertices of $C_{t,v,k,\lambda}$ correspond to the minimum coverings. In addition, the three polyhedra are nicely related by $P_{t,v,k,\lambda} \cap C_{t,v,k,\lambda} = T_{t,v,k,\lambda}$.

Our study of design polyhedra was motivated by two potential applications. The first one was the use of facets of these polyhedra in algorithms that construct designs or decide they do not exist. The second one was the possibility of deriving theoretical results using the knowledge of these facets. In this thesis, we mainly explore the algorithmic use of facets. Nevertheless, we illustrate the second motivation in Section 1.3.5, by giving a polyhedral proof of a known result. Far from being a simpler proof, it does, however, point to a direction for further investigation.

We end this section by presenting an overview of the thesis, including a summary of the main contributions. In the remaining sections of this chapter, we give an introduction to polyhedral theory, and definitions and known results of the combinatorial design problems discussed in the thesis.

In Chapter 2, we investigate polytopes associated with design problems. We present integer programming formulations for designs. These models use tk-matrices, and were previously employed in other kinds of algorithms for t-designs. We extend these formulations to packings and coverings, and discuss alternative formulations. Then, we survey known results on related problems, namely set packing, set partitioning and set covering. After that, we present the main results of this chapter (Section 2.3). We study the design polytopes, especially the polytopes for packing designs and m-sparse triple systems. For general packing designs, two classes of inequalities are studied, namely the generalized clique and the subpacking inequalities. We give a characterization of generalized cliques for packing designs as intersecting families of sets. We then summarize the main consequences of our results on these families for the design problems. The proofs are involved and we give them in Chapter 3. One of the consequences of these results is an efficient separation algorithm for designs with $k = t + 1$. We derive a new class of inequalities, which we call subpacking inequalities. We give some conditions under which these inequalities do not define facets, and other conditions regarding the extension of subpacking facets of $P_{t,v,k,\lambda}$ to facets of $P_{t,v',k,\lambda}$, for $v' \geq v$. In particular, for $k = t + 1$, subpacking facets are always extendible. Moreover, we show some specific instances of facet defining subpacking inequalities for $k = 3$ and $t = 2$, which define facets of $P_{2,v,3,1}$ for arbitrary v. We also investigate t-designs and packings avoiding subconfigurations. More precisely, we study the polytope of the m-sparse triple systems. Avoidance properties can be naturally formulated as additional inequalities. We show that these inequalities induce facets for the polytope of the m-sparse triple systems. Finally, we relate null designs to adjacency in the $T_{t,v,k,1}$ polytope. We show that if the difference of two designs is a (t, k)-pod, a null design of minimum support, the designs are adjacent as vertices of the $T_{t,v,k,1}$ polytope. We conclude Chapter 2 by showing some design polytopes for small parameters. The facet inducing inequalities of these polytopes were found using PORTA software, a public domain software package for polyhedral manipulation [20].

Chapter 3 is devoted to the study of maximal s-wise t-intersecting families of sets. These set systems correspond to generalized clique facets for the packing design polytope, as previously mentioned. In addition, they are interesting objects in their own right, and have been given a lot of attention in extremal set theory. A family $\mathcal{A}$ of k-subsets of a v-set is said to be s-wise t-intersecting, if the intersection among any s of the subsets in the family has cardinality at least t; such a family is said to be maximal if it is not properly contained in any other family with the same parameters (s, t, k, v). The famous Erdös-Ko-Rado theorem [33] is concerned with the maximum size of such a family for $s = 2$. Several other extremal properties of these families have been studied (see [42]). The extremal problem we are concerned with is the classification of maximal (with respect to set inclusion) s-wise t-intersecting families of k-subsets of a v-set. This turns out to be a wide open problem in extremal set theory. Our main results in this chapter are as follows: (1) we solve the classification problem for the first nontrivial case, namely $s = 2$, $k = t + 2$ and arbitrary v; (2) we prove that the classification problem does not depend on v. We study properties relating generating sets, kernels and essential elements of maximal set systems. We use these properties to show that given (s, t, k), the classification problem is independent on v. Then, we give a construction for maximal families with parameters $(s, t + 1, k + 1, v + 1)$ using the ones with (s, t, k, v). We show that any family with $s = 2, k = t + 2, t \geq 3$ can be generated by this construction from a smaller family. Combining all previous results we reduce the classification of maximal families for $(2, t, t + 2, v)$ to the cases $(2, 1, 3, 7)$ and $(2, 2, 4, 8)$, which we enumerate by computer. The results in the chapter appear in article [77].

In Chapter 4, we propose a polyhedral algorithm for constructing t-designs and packings. The algorithm uses the well-known branch-and-cut approach [16]. We investigate the application of this method to design problems and describe our implementation. Separation algorithms are developed for clique inequalities and m-sparse inequalities for $m = 4, 5$. The difficulty of having several nodes in the branch-and-cut tree corresponding to isomorphic problems is addressed by a partial isomorph rejection algorithm. Computational experiments showing the interplay of various parameters are analyzed. Our experiments address issues such as a comparison between branch-and-cut and branch-and-bound techniques, the effectiveness of both the specialized clique separation and the partial isomorph rejection, as well as the effect of other parameters on the algorithm's performance. We compare our results to the ones obtained by a general purpose integer programming package, as a reference point. The experiments shed some light on the potential and difficulties of applying these methods to design problems, and provide a starting point for further research in this area. Besides reproducing known results, our algorithm constructs some maximal cyclic packings, which to the best of our knowledge have not been previously investigated. Finally, in Chapter 5, we discuss open problems and indicate directions for further research.

1.2 A first glance at polyhedral theory

In polyhedral combinatorics, polyhedra are usually given implicitly, and our objective is to find their explicit representations. In the situation we are interested in, a polyhedron is implicitly defined by having designs as its vertices. The goal is to determine the linear inequalities that describe this polyhedron, so that we can compute its vertices, i.e., find the corresponding designs.

In this section, we list some classical results in polyhedral theory and outline their relation to linear programming and combinatorial optimization. Complete treatments of polyhedral theory can be found in the survey papers by Bachem and Grötschel [6] and Pulleyblank [86], and in the book by Nemhauser and Wolsey [80].

1.2.1 Basic polyhedral theory and linear programming

The proofs for the results contained in this subsection can be found in [80]. Throughout this subsection, let n and m be positive integers. A *polyhedron* $P \subseteq \mathbb{R}^n$ is the set of points satisfying a finite set of linear inequalities; that is, for any matrix $A \in \mathbb{R}^{m \times n}$ and vector $b \in \mathbb{R}^n$, $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is a polyhedron. A *polytope* is a bounded polyhedron, i.e., a polyhedron $P \subseteq \mathbb{R}^n$ is a polytope if and only if there exist vectors $l, u \in \mathbb{R}^n$ such that $l \leq x \leq u$ for all $x \in P$.

In a *linear programming problem*, we are interested in maximizing (minimizing) a linear function over a polyhedron. Let $c \in \mathbb{R}^n$ be the profit (cost) vector and $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ be the associated polyhedron. We can write the linear programming problem as

$$\text{(LP)} \quad \begin{cases} \max \ c^T x \\ Ax \leq b. \end{cases}$$

Minimization problems, as well as problems with different inequality signs ($=$,$\geq$), can be easily brought to the form given above by simple algebraic manipulations. The study of polyhedra is fundamental for linear programming.

First, we review some basic definitions from linear algebra; then, we define the concepts of dimension, faces, and facets of a polyhedron; finally, we state classical results on minimal representations of polyhedra.

A point x is called a *linear combination* of $x^1, \ldots, x^k \in \mathbb{R}^n$ if there exist $\alpha = (\alpha_1, \ldots, \alpha_k) \in \mathbb{R}^k$ such that $x = \sum_{i=1}^k \alpha_i x^i$. If $\alpha \geq \mathbf{0}$ then x is a *conic combination*. If $\sum_{i=1}^k \alpha_i = 0$ then x is an *affine combination*. If $\alpha \geq \mathbf{0}$ and $\sum_{i=1}^k \alpha_i = 1$ then x is a *convex combination*.

Given a nonempty subset $S \subseteq \mathbb{R}^n$, the set of all linear (conic, affine, convex) combinations of points in S is called the *linear (conic, affine, convex) hull* of elements in S and is denoted by *lin(S)* (*cone(S)*, *aff(S)*, *conv(S)*). We say that S is a *cone* if $S = cone(S)$ and that it is a *convex set* if $S = conv(S)$. It is easy to prove that any polyhedron is a convex set, and that a polyhedron of the form $\{x \in \mathbb{R}^n : Ax \leq \mathbf{0}\}$ is a cone.

A set of points $x^1, \ldots, x^k \in \mathbb{R}^n$ is *linearly independent* if the unique solution $\alpha \in \mathbb{R}^k$ to $\sum_{i=1}^k \alpha_i x^i = 0$ is $\alpha_i = 0$ for $i = 1, \ldots, k$. Analogously, a set of points $x^1, \ldots, x^k \in \mathbb{R}^n$ is *affinely independent* if the unique solution $\alpha \in \mathbb{R}^k$ to $\sum_{i=1}^k \alpha_i x^i = 0$, $\sum_{i=1}^k \alpha_i = 0$ is $\alpha_i = 0$ for $i = 1, \ldots, k$. The following proposition relates both concepts.

Proposition 1.2.1 *The following statements are equivalent:*

i. The set of points $x^1, \ldots, x^k \in \mathbb{R}^n$ is affinely independent.

ii. The set of points $x^2 - x^1, \ldots, x^k - x^1 \in \mathbb{R}^n$ is linearly independent.

iii. The set of points $(x^1, -1), \ldots, (x^k, -1) \in \mathbb{R}^{n+1}$ is linearly independent.

Since affine independence of points in a polyhedron is invariant under translation, affine independence is more suitable than linear independence for dealing with polyhedra.

Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. It is convenient to make some distinction among inequalities that define a polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq b\}$. Let $I = \{1, \ldots, m\}$, $I^= = \{i \in I : a^i x = b_i \text{ for all } x \in P\}$ and let $I^{\leq} = \{i \in I : a^i x < b_i \text{ for some } x \in P\} = I \setminus I^=$. We indicate by $(A^=, b^=)$ and $(A^{\leq}, b^{\leq})$ the rows of (A, b) corresponding to $I^=$ and $I^{\leq}$, respectively, the equality and inequality representations of P with respect to (A, b).

A polyhedron $P \subseteq \mathbb{R}^n$ is of *dimension* k, denoted by $dimP = k$, if the maximum number of affinely independent points in P is $k + 1$. We say that P is *full dimensional* if $dimP = n$. The maximum number of linear independent rows of a matrix $A \in \mathbb{R}^{m\times n}$ is the *rank* of A, denoted by $rank(A)$.

Proposition 1.2.2 *If $P \subseteq \mathbb{R}^n$ then $dimP + rank(A^=, b^=) = n$.*

Let $d \in \mathbb{R}^n$ and $d_0 \in \mathbb{R}$. An inequality $d^T x \leq d_0$ is said to be *valid* for P if it is satisfied by all points of P. A subset $F \subseteq P$ is called a *face* of P if there exists a valid inequality $d^T x \leq d_0$ such that $F = P \cap \{x \in \mathbb{R}^n : d^T x = d_0\}$. The inequality is said to *represent* or to *induce* the face F. It is easy to see that P and $\emptyset$ are faces of P, and that a face of a polyhedron is itself a polyhedron. A face F of P is called *proper* if $F \neq P$ and *nontrivial* if $F \neq \emptyset$. A maximal nontrivial proper face F of P is called a *facet* of P. A face of dimension zero is called a *vertex* and a face of dimension one is called an *edge* of P. A polyhedron that has at least one vertex is said to be *pointed.* Let $P^0 = \{x \in \mathbb{R}^n : Ax \leq \mathbf{0}\}$ be the polyhedral cone associated with $P = \{x \in \mathbb{R}^n : Ax \leq b\}$. Any point $r \in P^0$ is called a *ray* of P and if r belongs to an edge of P^0 it is called an *extreme ray* of P.

The study of faces is important for linear programming. If a linear programming problem, $\max c^T x$ subject to $x \in P$, has an optimal solution, it is easy to see that the set of all optimal solutions is a face of the polyhedron P. Just note that if $c_0 = \max\{c^T x : x \in P\}$, then $c^T x \leq c_0$ is a valid inequality for P that induces the face $F = \{x \in P : c^T x = c_0\}$, which is the set of optimal solutions to the problem.

The main result concerning minimal inequality representation is summarized below.

Theorem 1.2.3 *Let P be a polyhedron.*

i. If P is full dimensional then it has a unique (up to scalar multiplication) minimal representation by a finite set of linear inequalities, each one representing each of the facets of P.

ii. If $dim P = n - k$ with $k > 0$ then a minimal representation of P is given by $\{x \in \mathbb{R}^n : a^i x = b_i,\ i = 1, \ldots, k,\ a^i x \leq b_i,\ i = k+1, \ldots k+t\}$, where $\{(a^i, b_i)\}_{i=1,\ldots,k}$ is a maximal set of linearly independent rows of $(A^=, b^=)$, and for $i = k+1, \ldots k+t$, $a^i x \leq b_i$ are inequalities representing each one of the facets F_i of P.

We now give two characterizations of facets that are very useful in deciding whether or not an inequality is facet inducing.

Theorem 1.2.4 *Let F be a nonempty proper face of $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ and $(A^=, b^=)$ be the equality representation of P. Then, the following statements are equivalent:*

i. F is a facet of P.

ii. $dim F = dim P - 1$.

iii. For any pair of inequalities $d^T x \leq d_o$ and $\overline{d}^T x \leq \overline{d}_o$ that are valid for P and represent F, there exist $\alpha \in \mathbb{R}^m$ and $\gamma > 0 \in \mathbb{R}$ such that

$$d = \gamma \overline{d} + \alpha^T A^=, \quad d_o = \gamma \overline{d}_o + \alpha^T b^=.$$

By now we have seen one of the representations of a polyhedron, namely the one by a minimum number of inequalities. Another way to represent a polyhedron is by means of its vertices and extreme rays, as stated in the following theorem.

Theorem 1.2.5 *(Minkowski's theorem) Let P be a pointed polyhedron, and P_0 be its associated polyhedral cone. Then*

i. P has a finite set of vertices V,

ii. there is a finite set of extreme rays R of P such that $cone(R) = P_0$, and *iii.* $P = conv(V) + cone(R)$.

Remark: Most polyhedra we will deal with are polytopes, i.e. bounded polyhedra. If P is a polytope then $P = conv(V)$, since for polytopes $cone(R) = \{\mathbf{0}\}$.
The previous theorem is the basis for the simplex algorithm for linear programming. The simplex method also uses the fact that if there is an optimal solution for a linear programming problem over a pointed polyhedron P, then there is an optimal solution that is a vertex, in which case the algorithm returns an optimal vertex.

1.2.2 Combinatorial optimization and polyhedral methods

We describe a general combinatorial optimization problem. Let $E = \{e_1, \ldots, e_n\}$ be a ground set, c_i be a weight associated with $e_i \in E$ for $i = 1, \ldots, n$, and $\mathcal{F}$ be a family of feasible subsets. We want to find $S \in \mathcal{F}$ such that $c(S) = \sum_{i:e_i \in S} c_i$ is maximized. A natural representation of an element $S \in \mathcal{F}$ is by a 0-1 incidence vector, that is a vector $x^S \in \mathbb{R}^n$ such that $x_i^S = 1$ if and only if $e_i \in S$. This gives a polytope representation of the problem, namely the polytope $C = conv\{x^S : S \in \mathcal{F}\}$. This polytope has the nice property of its vertices being members of $\mathcal{F}$, as we have seen for the particular case of designs. Moreover, solving the combinatorial problem is equivalent to finding an optimal vertex of this polytope, which can be done via linear programming methods. The difficulty lies in the fact that we need a description of this polytope in terms of linear inequalities.

Sometimes a combinatorial optimization problem is given in the form of an integer programming problem. An integer programming problem is one of the form

$$\text{(IP)} \quad \begin{cases} \max\ c^T x \\ Ax \leq b \\ x \text{ integer.} \end{cases}$$

Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$. The polyhedral representation of this problem is given by $P_I = conv\{x \in P : x \text{ is integer}\}$. We would like to obtain a complete representation of P_I by means of facet inducing inequalities. A useful technique for finding such inequalities is given by the *Chvátal-Gomory* cut generation. Let $\alpha \geq \mathbf{0} \in \mathbb{R}^n$. Then $(\alpha^T A)x \leq \alpha^T b$ is a valid inequality for P_I implied by $Ax \leq b$. If $\alpha^T A$ is integral then any integer solution x to the system will make $(\alpha^T A)x$ integral, so the inequality $(\alpha^T A)x \leq \lfloor \alpha^T b \rfloor$ is valid for P_I. Such an inequality is called a *Chvátal-Gomory cut* derived from $Ax \leq b$.

The following theorem was proven by Chvátal [21] for polytopes, and generalized by Schrijver [91] for polyhedra.

Theorem 1.2.6 *Let P and P_I be the polyhedra given by an integer programming problem as defined previously. Let $P^0 \supseteq \cdots \supseteq P^i \supseteq P^{i+1} \supseteq \cdots \supseteq P_I$ be a sequence of polyhedra recursively defined by: $P^0 = P$, P^{i+1} is the polyhedron obtained by the inequalities defining P^i plus all the Chvátal-Gomory cuts derived from these inequalities, for $i \geq 0$. Then, there is an integer t such that $P^t = P_I$, i.e., the sequence is finite.*

The main consequence of this theorem is that any facet inducing inequality can be obtained by a finite number of applications of the Chvátal-Gomory procedure. Chvátal-Gomory cuts have been used to derive and prove the validity of inequalities for polyhedra of structured problems. They can also be seen as a technique for generating valid and facet inducing inequalities.

1.3 t-designs, packings and coverings

In this section, we survey general results on t–(v,k,λ) designs, t–(v,k,λ) packings and t–(v,k,λ) coverings. First, we define related designs, then we review results on the existence of t-designs and on the determination of packing and covering numbers, and finally we describe some problems on Steiner triple systems avoiding subconfigurations.

1.3.1 Balanced incomplete block designs and Steiner systems

Several well-studied designs are special cases of t-designs, or to be historically accurate, t-designs are generalizations of pairwise balanced designs, first introduced in statistical applications.

A balanced incomplete block design BIBD(v,b,r,k,λ) is a 2–(v,k,λ) design; the parameters b and r are determined by the other three and stand for, respectively, the

A 2–$(6,3,2)$ design:	$\{1,2,3\},\{1,2,4\},\{1,3,5\},\{1,4,6\},\{1,5,6\},$ $\{2,3,6\},\{2,4,5\},\{2,5,6\},\{3,4,5\},\{3,4,6\}.$
A 3–$(8,4,1)$ design:	$\{1,2,3,4\},\{1,2,5,6\},\{1,2,7,8\},\{1,3,5,7\},\{1,3,6,8\},$ $\{1,4,5,8\},\{1,4,6,7\},\{2,3,5,8\},\{2,3,6,7\},\{2,4,5,7\},$ $\{2,4,6,8\},\{3,4,5,6\},\{3,4,7,8\},\{5,6,7,8\}.$

Table 1.1: Examples of t-designs.

number of blocks and the replication of points (i.e. the number of blocks that contain any particular point). These designs are also referred to as (v,k,λ) BIBDs.

A Steiner system $S(t,k,v)$ is a t–$(v,k,1)$ design. A Steiner triple system $STS(v)$ is a 2–$(v,3,1)$ design, and a Steiner quadruple system $SQS(v)$ is a 3-$(v,4,1)$ design. Sometimes t–(v,k,λ) designs are referred to as $S_\lambda(t,k,v)$.

1.3.2 Existence results for t-designs

It can be shown by a counting argument that for any $0 \le s \le t$, a t–(v,k,λ) design is an s–(v,k,λ_s) design with

$$\lambda_s = \frac{\lambda\binom{v-s}{t-s}}{\binom{k-s}{t-s}}. \tag{1.1}$$

In particular, λ_s must be an integer, for all $0 \le s \le t$. Let b be the number of blocks and r be the replication number in a t–(v,k,λ) design. Then, clearly $b = \lambda_0$ and $r = \lambda_1$.

Equation (1.1) gives a necessary condition for the existence of a t–(v,k,λ) design made explicit as follows. For any t–(v,k,λ) design we must have

$$\lambda\binom{v-s}{t-s} \equiv 0 \quad \left(\mathrm{mod}\binom{k-s}{t-s}\right). \tag{1.2}$$

for all $0 \le s \le t$. We say that a t–(v,k,λ) design is *admissible* if its parameters satisfy the previous conditions.

Another necessary condition is given by the following theorem, which is a generalization of Fisher's inequality.

Theorem 1.3.1 *(Fisher's inequality for general t - Ray-Chaudhuri & Wilson [87])*
Let $t < k \leq v - t$. If $(V, \mathcal{B})$ is a simple t–(v, k, λ) design, then $|\mathcal{B}| \geq \binom{v}{\lfloor t/2 \rfloor}$.

Wilson showed that the necessary conditions given in (1.2) are sufficient for λ large enough.

Theorem 1.3.2 *(Wilson [104])*
Given $0 < t \leq k \leq v$, then there exist an integer λ^0, such that any admissible t–(v, k, λ) design (possibly with repeated blocks) exists for any $\lambda \geq \lambda^0$.

The following theorem guarantees the existence of simple (i.e. with no repeated blocks) t-designs for arbitrarily large t. However, the values of λ are extremely large.

Theorem 1.3.3 *(Teirlinck [99])*
Given $v > t \geq 0$ with $v \equiv t \pmod{(t+1)!^{2t+1}}$, then a simple t–$(v, t+1, (t+1)!^{2t+1})$ exists.

The two previous theorems give no light on the existence of simple t-designs with small λ. We summarize existence results for such cases.

Let us start with Steiner systems or t-designs with $\lambda = 1$. A $STS(v)$ exists if and only if $v \equiv 1, 3 \pmod 6$. A $SQS(v)$ exists if and only if $v \equiv 2, 4 \pmod 6$. Only a finite number of $S(t, k, v)$ with $t = 4, 5$ are known and none are known for $t \geq 6$.

Let us discuss the 2-designs or BIBDs (of course, some of these results include $S(2, k, v)$ designs). For $k = 3, 4, 5$ all admissible 2-(v, k, λ) designs exist with the exception of a 2-$(15, 5, 2)$ design. The existence of 2–$(v, 6, \lambda)$ is still unsettled, with the smallest unsolved case being 2–$(46, 6, 1)$.

For t-designs with $t \geq 3$ and small λ the situation is far from settled. There are many examples and some infinite families of simple t-designs with $t = 4, 5, 6$, but these infinite families have growing λ. Recently, many 7-designs (with $\lambda = 4$ and larger) and a few 8-designs have been found [12].

For a more precise account of the existence situation in general and for specific parameters, we refer the reader to the comprehensive collection of results, pointers to results and tables, given in the handbook edited by Colbourn and Dinitz [24], in particular chapters [1] [25] [57] [72].

1.3.3 Determination of packing and covering numbers

In this section, we outline some general results on the determination of the size of maximum packings and minimum coverings. For a comprehensive account of results the reader is referred to the survey papers of Mills and Mullin [75] and Stinson [95] [96].

The following theorems provide upper bounds for the maximum size of a packing $D_\lambda(v,k,t)$ and lower bounds for the minimum size of a covering $C_\lambda(v,k,t)$. First, we comment on the relation of these quantities and similar ones for simple designs, which are denoted by $\overline{D}_\lambda(v,k,t)$ and $\overline{C}_\lambda(v,k,t)$. Note that that the upper bounds on $D_\lambda(v,k,t)$ and lower bounds on $C_\lambda(v,k,t)$ are also bounds for $\overline{D}_\lambda(v,k,t)$ and $\overline{C}_\lambda(v,k,t)$, since

$$\overline{D}_\lambda(v,k,t) \leq D_\lambda(v,k,t) \leq C_\lambda(v,k,t) \leq \overline{C}_\lambda(v,k,t).$$

Moreover, both packings and minimum coverings for $\lambda = 1$ cannot have repeated blocks, thus $\overline{D}_1(v,k,t) = D_1(v,k,t)$ and $\overline{C}_1(v,k,t) = C_1(v,k,t)$.

Theorem 1.3.4 *(Schönheim bounds)*
The following bound holds $D_\lambda(v,k,t) \leq \left\lfloor \frac{v}{k}\, D_\lambda(v-1,k-1,t-1) \right\rfloor$. Iterating this bound gives

$$D_\lambda(v,k,t) \leq U_\lambda(v,k,t) := \left\lfloor \frac{v}{k} \left\lfloor \frac{v-1}{k-1} \cdots \left\lfloor \frac{v-t+1}{k-t+1}\lambda \right\rfloor \cdots \right\rfloor \right\rfloor.$$

Moreover, if $\lambda(v-1) \equiv 0 \pmod{(k-1)}$ and $\lambda v(v-1) \equiv -1 \pmod{k}$, then

$$D_\lambda(v,k,2) \leq U_\lambda(v,k,2) - 1.$$

Analogously, we have $C_\lambda(v,k,t) \leq \left\lceil \frac{v}{k}\, C_\lambda(v-1,k-1,t-1) \right\rceil$. Iterating this bound gives

$$C_\lambda(v,k,t) \geq L_\lambda(v,k,t) := \left\lceil \frac{v}{k} \left\lceil \frac{v-1}{k-1} \cdots \left\lceil \frac{v-t+1}{k-t+1}\lambda \right\rceil \cdots \right\rceil \right\rceil.$$

Moreover, if $\lambda(v-1) \equiv 0 \pmod{(k-1)}$ and $\lambda v(v-1) \equiv 1 \pmod{k}$, then

$$C_\lambda(v,k,2) \geq L_\lambda(v,k,2) + 1.$$

Observe that, if a t-(v,k,λ) design exists it is a t-(v,k,λ) packing and a t-(v,k,λ) covering. In addition, letting b be the number of blocks in the t–(v,k,λ) design, the divisibility conditions given in (1.2) imply that $U_\lambda(v,k,t) = L_\lambda(v,k,t) = b = \frac{\lambda\binom{v}{t}}{\binom{k}{t}}$, which implies that a t-design is both a maximum packing and a minimum covering.

A 2–(5, 3, 1) maximum packing:	A 2–(5, 3, 1) minimum covering:
$\{1,2,3\},\{1,4,5\}$.	$\{1,2,3\},\{1,4,5\},\{2,4,5\},\{3,4,5\}$.
$U_1(5,3,2) = \left\lfloor \frac{5}{3} \left\lfloor \frac{4}{2}\, 1 \right\rfloor \right\rfloor = 3$	$L_1(5,3,2) = \left\lceil \frac{5}{3} \left\lceil \frac{4}{2}\, 1 \right\rceil \right\rceil = 4$
$D_1(5,3,2) = U_1(5,3,2) - 1 = 2$	$C_1(5,3,2) = L_1(5,3,2) = 4$

Table 1.2: Example of a packing and a covering meeting the Schönheim bounds.

Theorem 1.3.5 *(Johnson bound for packings) If $v < \frac{k^2}{(t-1)}$, then*

$$D_\lambda(v,k,t) \le \left\lceil \frac{(k-t+1)}{k^2-(t-1)v} \right\rceil .$$

We now mention some results on the determination of packing and covering numbers (see [75] [95] [96]). These numbers are obtained through constructions of packings and coverings attaining known bounds. The packing numbers $D_\lambda(v,3,2)$ and $D_\lambda(v,4,2)$ have been completely determined. The packing numbers $D_1(v,4,3)$ for $v \not\equiv 5 \pmod 6$ have also been obtained. Partial results are known for $D_\lambda(v,5,2)$. The covering numbers $C_\lambda(v,3,2)$ and $C_1(v,4,2)$ have been completely determined. In addition, the covering numbers $C_1(v,4,3)$ for $v \not\equiv 7 \pmod{12}$ have been obtained. Partial results are known for $C_\lambda(v,5,2)$.

1.3.4 Steiner triple systems avoiding subconfigurations

As we have seen in the previous two sections, the basic questions regarding triple systems (i.e., 2-$(v,3,\lambda)$ designs, packings and coverings) have been settled. However, we find many open problems for triple systems satisfying some additional properties. In this section, we concentrate on Steiner triple systems (i.e. 2-$(v,3,1)$ designs) avoiding some forbidden configurations. The same constraints can be imposed on 2-$(v,3,1)$ packings and coverings, leading to relevant problems (for example, see [18] for an application of anti-Pasch 2-$(v,3,1)$ packings).

The reader is directed to the survey by Grannell and Griggs [45] for a detailed treatment of avoidance problems.

The unique $STS(7)$ contains Pasches:
$\{\mathbf{1,2,3}\}, \{\mathbf{1,4,5}\}, \{1,6,7\}, \{2,4,6\}, \{\mathbf{2,5,7}\}, \{\mathbf{3,4,7}\}, \{3,5,6\}$.
The unique $STS(9)$ is anti-Pasch:
$\{1,2,3\}, \{1,4,7\}, \{1,5,9\}, \{1,6,8\}, \{2,4,9\}, \{2,5,8\},$
$\{2,6,7\}, \{3,4,8\}, \{3,5,7\}, \{3,6,9\}, \{4,5,6\}, \{7,8,9\}$.

Table 1.3: Examples of STSs with and without Pasch configurations

A (p,l)-configuration in a Steiner triple system is a set of l blocks (of the Steiner triple system) spanning p elements. The *Pasch or quadrilateral configuration* is the unique $(6,4)$-configuration, which is of the form $\{\{a,b,c\},\{a,d,e\}, \{f,b,d\}, \{f,c,e\}\}$. A *mitre* is a $(7,5)$-configuration on 7 elements, say, a,b,c,d,e,f,g of the form $\{\{a,b,e\},\{a,c,f\}, \{a,d,g\},\{b,c,d\},\{e,f,g\}\}$. A STS is *anti-Pasch* (*anti-mitre*) if it does not contain a Pasch (mitre) as a subconfiguration.

Let $m \geq 4$. An $STS(v)$ is said to be m-sparse if it avoids every $(l+2,l)$-configuration for $4 \leq l \leq m$. Erdös (see [63]) conjectured that for all $m \geq 4$ there exists an integer v_m such that for every admissible $v \geq v_m$ there exist an m-sparse $STS(v)$.

The 4-sparse STSs are the same as anti-Pasch ones, since Pasches are the only $(6,4)$-configuration. A 4-sparse (or anti-Pasch) $STS(v)$ is known to exist for all $v \equiv 3 \pmod 6$ [14]. For the remaining case, i.e. the case $v \equiv 1 \pmod 6$, there are constructions and partial results which leads to 3/4 of the values being settled [14] [47] [48] [67] [88] [97]. Anti-mitre Steiner triple systems were first studied in [26]. The 5-*sparse* Steiner triple systems are the systems that are both anti-Pasch and anti-mitre. Although there are some results on 5-sparse STSs [26] [66], the problem is far from settled. As for $m \geq 6$, no m-sparse Steiner triple system is known.

1.3.5 A polyhedral proof for an old theorem

We illustrate how valid inequalities for the design polytope can be used for proving design properties, by giving a polyhedral proof of Theorem 1.3.4.

Theorem 1.3.6 *(Schönheim bounds)*

The packing number $D_\lambda(v,k,t)$ satisfies

$$D_\lambda(v,k,t) \leq U_\lambda(v,k,t) := \left\lfloor \frac{v}{k} \left\lfloor \frac{v-1}{k-1} \cdots \left\lfloor \frac{v-t+1}{k-t+1}\lambda \right\rfloor \cdots \right\rfloor \right\rfloor.$$

Similarly, for the covering number $C_\lambda(v,k,t)$ we have

$$C_\lambda(v,k,t) \geq L_\lambda(v,k,t) := \left\lceil \frac{v}{k} \left\lceil \frac{v-1}{k-1} \cdots \left\lceil \frac{v-t+1}{k-t+1}\lambda \right\rceil \cdots \right\rceil \right\rceil.$$

Proof. We show the inequality for the packing numbers; an analogous argument can be applied for the covering numbers.

Let $P_{t,v,k,\lambda}$ be the polytope for the $t-(v,k,\lambda)$ packing design. The inequalities

$$\sum_{K \in \binom{[1,v]}{k}:\ K \supseteq T} x_K \leq \lambda, \qquad \text{for all } T \in \binom{[1,v]}{t}, \tag{1.3}$$

are valid for $P_{t,v,k,\lambda}$. For simplicity, we will not repeat $K \in \binom{[1,v]}{k}$ in the next summations.

We shall derive the upper bound on $D_\lambda(v,k,t)$ by taking linear combinations of inequalities in (1.3). Let $S(\ell)$ be the following statement: "For any $L \subseteq [1,v]$, $|L| = t-\ell$, the inequality

$$\sum_{K \supseteq L} x_K \;\leq\; \left\lfloor \frac{v-(t-\ell)}{k-(t-\ell)} \left\lfloor \frac{v-(t-\ell+1)}{k-(t-\ell+1)} \cdots \left\lfloor \frac{v-(t-1)}{k-(t-1)}\lambda \right\rfloor \cdots \right\rfloor \right\rfloor \tag{1.4}$$

is valid for $P_{t,v,k,\lambda}$." We prove that $S(\ell)$ holds for $1 \leq \ell \leq t$ by induction on ℓ.

Consider the case $\ell = 1$. Let $L \subseteq [1,v]$ with $|L| = t-1$. For each t-subset $T \supseteq L$, we add inequalities in (1.3), obtaining

$$\sum_{T \supseteq L:\ |T|=t} \sum_{K \supseteq T} x_K \leq \sum_{T \supseteq L:\ |T|=t} \lambda.$$

Reorganizing the left-hand side and observing that $v-(t-1)$ inequalities were added, we get

$$\sum_{T \supseteq L:\ |T|=t} \left(\sum_{K \supseteq T, K \supseteq L} x_K + \sum_{K \supseteq T, K \not\supseteq L} x_K \right) \leq (v-(t-1))\lambda.$$

This implies

$$\sum_{T \supseteq L:\ |T|=t} \sum_{K \supseteq T, K \supseteq L} x_K \leq (v-(t-1))\lambda,$$

and rearranging sums, we obtain

$$\sum_{K \supseteq L} (k-(t-1))x_K \leq (v-(t-1))\lambda.$$

The Chvátal-Gomory cut corresponding to this inequality is

$$\sum_{K \supseteq L} x_K \leq \left\lfloor \frac{v-(t-1)\lambda}{k-(t-1)} \right\rfloor,$$

which implies $S(1)$.

Assume now that $S(\ell)$ is true; we shall prove $S(\ell+1)$. Let $L \subseteq [1,v]$ with $|L| = t-(\ell+1)$. Then, there are $v-(t-(\ell+1))$ sets M of cardinality $t-\ell$ containing L. By induction, inequality (1.4) is valid for any such M, and adding these inequalities gives

$$\begin{aligned} \sum_{M \supseteq L:\ |M|=t-\ell} \sum_{K \supseteq M} x_K \quad \leq \\ \sum_{M \supseteq L:\ |M|=t-\ell} \left\lfloor \frac{v-(t-\ell)}{k-(t-\ell)} \left\lfloor \frac{v-(t-\ell+1)}{k-(t-\ell+1)} \cdots \left\lfloor \frac{v-(t-1)}{k-(t-1)}\lambda \right\rfloor \cdots \right\rfloor \right\rfloor. \end{aligned}$$

Using an algebraic manipulation similar to the base case, we get

$$\begin{aligned} \sum_{K \supseteq L} (k-(t-(\ell+1)))x_K \quad \leq \\ (v-(t-(\ell+1))) \left\lfloor \frac{v-(t-\ell)}{k-(t-\ell)} \left\lfloor \frac{v-(t-\ell+1)}{k-(t-\ell+1)} \cdots \left\lfloor \frac{v-(t-1)}{k-(t-1)}\lambda \right\rfloor \cdots \right\rfloor \right\rfloor. \end{aligned}$$

Again, taking the Chvátal-Gomory cut corresponding to the previous inequality completes the proof for $S(\ell+1)$.

Finally, since $S(t+1)$ is valid for $P_{t,v,k,\lambda}$ it is satisfied in particular by the incidence vector of a maximum packing, which completes the proof. □

In the previous proof we have used valid inequalities of the form (1.3). The method is quite general and other inequalities could be used in place of these ones.

Chapter 2

Polytopes for Designs

Viewing designs as integer solutions to systems of equations has contributed to the study of t-designs in several ways. It involves tk-matrices and their associated systems of equations (DP) that we define in Section 2.1.

System (DP) was used by Wilson [104] to show that the necessary conditions for the existence of t–(v, k, λ) designs are sufficient for λ large enough (see Theorem 1.3.2). This was done by showing that (DP) admits nonnegative integer solutions for large λ. Note that he required the x_i's to be nonnegative integers, rather than in $\{0, 1\}$. We direct the reader to the notes "Linear algebra and designs" by Godsil [44] for a detailed study of tk-matrices, their properties and use in proving results on t-designs.

Another remarkable use of system (DP) led to the discovery of several previously unknown simple designs for $t \geq 4$ and small λ. Kramer and Mesner [56] first observed how system (DP) can be transformed into system ($\mathrm{DP_A}$) (see Section 2.1.1) in order to deal with designs with prescribed automorphism groups. The transformation often leads to a significant reduction on the problem sizes. Solving systems of the form ($\mathrm{DP_A}$) yielded several new results, including the first three examples of 5-designs with an odd number of points by Kramer [54] and Magliveras and Leavitt [69], and several other t-designs for $t = 4, 5, 6$ by Kreher and Radziszowski [58] [59] [60], including the existence of a 6–$(14, 7, 4)$ design, the smallest nontrivial admissible 6-design. The latter authors applied the lattice basis reduction algorithm proposed in [64] to solve ($\mathrm{DP_A}$). Schmalz [90] proposed an algorithm to compute the isomorphism types or even the designs themselves, finding new 6-designs, using the same system of equations. Betten, Laue and Wassermann [12] employed the same equations to find several 7 and 8-designs.

In this thesis we propose the study of the polytopes associated with solutions of the systems (DP) and ($\mathrm{DP_A}$), and of their extension to systems for packing and covering designs (see Section 2.1). These polytopes produce "tighter", more restrictive, systems of inequalities. The success obtained by applying polyhedral algorithms to other combinatorial optimization problems [84] (also see [16]) was an incentive to pursue this investigation, which led to the algorithms in Chapter 4. Moreover, we suggest a still unexplored potential use of design polytopes in the theory of combinatorial designs. It is our hope that some of the new inequalities and facets discussed in Section 2.2 and Section 2.3 can be used to derive new results on the existence of t-designs and/or new bounds for packings and coverings. Theorem 1.3.6 illustrates how the inequalities in problem (PDP) can be used to derive the Schönheim bounds for packings.

The rest of this chapter is organized as follows. In Section 2.1, design problems are formulated as integer programming problems. This section includes the formulation (DP) mentioned earlier as well as its extension for dealing with packings and coverings. These formulations are special cases of well-known combinatorial optimization problems, namely set packing, set partitioning, and set covering. In Section 2.2, polyhedral studies of these general problems are revisited, leading as corollaries to results for design polytopes. The main contribution of this chapter is contained in Section 2.3. In this section, we study the polytopes for designs. After some basic results for the three design polytopes, two classes of inequalities for the packing design polytope are studied, other facets for m-sparse triple systems are developed, separation results for the mentioned inequalities are shown and adjacency results for the t-design polytope are included. In the last section of the chapter, Section 2.4, the reader can find some samples of small polytopes for which the complete description by linear inequalities is available.

Parts of several sections of this chapter appeared in [76]; however, most of the main results of the chapter (Section 2.3) are first shown here.

In this chapter, we state when some facets are polynomial-time separable. The size of a design problem is measured by the number of bits needed for its integer programming formulation. For t–(v,k,λ) designs, the problem size is exactly $\binom{v}{t} \times \binom{v}{k} + \binom{v}{t} \log \lambda$. Our measure of complexity is the number of basic operations such as arithmetic operations and comparisons.

2.1 Integer programming formulations for design problems and tk-matrices

In this section, we present integer programming formulations for t-designs and their extensions to packings and coverings. Similar formulations for designs with prescribed automorphism groups are shown in Section 2.1.1. Other integer programming formulations are shown in Section 2.1.2. Although they are not used in this thesis, they offer alternative formulations to some of the problems discussed here.

First, we define a tk-matrix, which encodes the incidence between t-subsets and k-subsets of a v-set. More precisely, let $W^v_{t,k}$ be the $\binom{v}{t} \times \binom{v}{k}$ matrix with rows indexed by the t-subsets and columns by k-subsets of a v-set

$$\left[W^v_{t,k}\right]_{T,K} = \begin{cases} 1 & \text{if } T \subseteq K, \\ 0 & \text{otherwise.} \end{cases}$$

For a detailed study of these matrices and their role in design theory see [44]. It is easy to see that t–(v,k,λ) designs correspond to the solutions $x \in \mathbb{R}^{\binom{v}{k}}$ of

$$(\mathrm{DP}) \quad \begin{cases} W^v_{t,k}\ x = \lambda \mathbf{1}, \\ x \in \{0,1\}^{\binom{v}{k}}. \end{cases}$$

The maximum packings correspond to solutions $x \in \mathbb{R}^{\binom{v}{k}}$ of

$$(\mathrm{PDP}) \quad \begin{cases} \text{maximize} & \mathbf{1}^T x \\ \text{subject to} & W^v_{t,k}\ x \leq \lambda \mathbf{1}, \\ & x \in \{0,1\}^{\binom{v}{k}}. \end{cases}$$

Analogously, the minimum coverings correspond to solutions $x \in \mathbb{R}^{\binom{v}{k}}$ of

$$(\mathrm{CDP}) \quad \begin{cases} \text{minimize} & \mathbf{1}^T x \\ \text{subject to} & W^v_{t,k}\ x \geq \lambda \mathbf{1}, \\ & x \in \{0,1\}^{\binom{v}{k}}. \end{cases}$$

For designs with possibly repeated blocks, the condition "$x \in \{0,1\}^{\binom{v}{k}}$" should be replaced by "x nonnegative integer", or alternatively, the matrix $W_{t,k}^{v}$ should be replaced by the same matrix with each column repeated λ times and the points x considered to be in $\{0,1\}^{\lambda\binom{v}{k}}$.

Based on these integer programming formulations, we rewrite the design polytopes defined in Chapter 1 as

$$\begin{aligned} T_{t,v,k,\lambda} &= conv\{x \in \{0,1\}^{\binom{v}{k}} : W_{t,k}^{v}\ x = \lambda\mathbf{1}\}, \\ P_{t,v,k,\lambda} &= conv\{x \in \{0,1\}^{\binom{v}{k}} : W_{t,k}^{v}\ x \leq \lambda\mathbf{1}\}, \\ C_{t,v,k,\lambda} &= conv\{x \in \{0,1\}^{\binom{v}{k}} : W_{t,k}^{v}\ x \geq \lambda\mathbf{1}\}. \end{aligned}$$

For $\lambda = 1$ these are special cases of set partitioning, set packing and set covering polytopes, respectively. For $\lambda > 1$, the polytopes $P_{t,v,k,\lambda}$ and $C_{t,v,k,\lambda}$ can be brought to the form of the polytope for independent sets of an independence system (see Section 2.2 and Section 2.3).

The following proposition relates the problems just described. It can be proven using the Schönheim bounds (see Section 1.3.3), but we give an alternative proof.

Proposition 2.1.1 *Assume that a t–(v,k,λ) design exists. Let $x^* \in \mathbb{R}^{\binom{v}{k}}$. Then the following statements are equivalent:*

i. x^* is a solution to (DP).

ii. x^* is an optimal solution to (PDP).

iii. x^* is an optimal solution to (CDP).

Proof. Let $b = \lambda\binom{v}{t}/\binom{k}{t}$, the number of blocks in a t–(v,k,λ) design. Let x be any feasible solution to (PDP). Adding the inequalities of $W_{t,k}^{v}\ x \leq \lambda\mathbf{1}$, we get that $\binom{k}{t}\sum x_i \leq \binom{v}{t}\lambda$, that is, $\mathbf{1}^T x \leq \binom{v}{t}\lambda/\binom{k}{t} = b$. Analogously, for any feasible solution y of (CDP), we see that $\mathbf{1}^T y \geq b$. By the existence of a t–(v,k,λ) design, we conclude that for any optimal solution x^* of either (PDP) or (CDP) we must have $\mathbf{1}^T x^* = b$. But this equality holds if and only if $W_{t,k}^{v}\ x^* = \lambda\mathbf{1}$. □

Proposition 2.1.1 suggests that any of the three formulations may be used to find t-designs.

2.1.1 Designs with prescribed automorphism groups

There are two main reasons why it is desirable to search for designs with a prescribed automorphism group. Classes of designs admitting certain automorphism groups are of special interest, such as cyclic and rotational designs. In addition, assuming the action of an automorphism group on a design allows for a reduction on the size of the corresponding integer programming problem.

Let V be a finite set with $|V| = v$. The set S_V of all permutations on V forms a symmetric group where composition is the group operation. If $\pi \in S_V$ maps the design $D_1 = (V, \mathcal{B}_1)$ to the design $D_2 = (V, \mathcal{B}_2)$ then we say that D_1 and D_2 are *isomorphic*. If π fixes D_1 so that $\mathcal{B}_1 = \mathcal{B}_2$ then π is called an *automorphism* of D_1. The set of all automorphisms of a design D, denoted by $AutD$, forms a group under the composition operation.

Let A be a permutation group acting on a set V. If $x \in V$, the *orbit* of x under A is defined by $A(x) = \{y \in V : \pi(x) = y \text{ for some } \pi \in A\}$. An equivalence relation on V can be defined by x and y being equivalent if and only if $y \in A(x)$; the equivalent classes are the different orbits of V. This definition is naturally extended to orbits of s-subsets of V. The orbit of an s-subset $S \in \binom{V}{s}$ is defined by $A(S) = \left\{T \in \binom{V}{s} : \pi(S) = T \text{ for some } \pi \in A\right\}$.

Let t, v and k be integers with $0 < t \leq k \leq v$, V be a set with $|V| = v$, and A be a permutation group acting on V. Let T_i, $i = 1, \ldots, \left|\binom{V}{t}/A\right|$, be representatives of the orbits of the t-subsets and K_j, $j = 1, \ldots, \left|\binom{V}{k}/A\right|$, be representatives of the orbits of the k-subsets. We define $W^A_{t,k}$ as the $\left|\binom{V}{t}/A\right| \times \left|\binom{V}{k}/A\right|$ matrix defined by

$$\left[W^A_{t,k}\right]_{i,j} = |\{K \in A(K_j) : K \supseteq T_i\}|.$$

Theorem 2.1.2 *(Kramer and Mesner [56]) Let A be a permutation group and $l = \left|\binom{V}{k}/A\right|$. The simple t-(v, k, λ) designs admitting an automorphism group A are the solutions x of*

$$(\mathrm{DP_A}) \quad \begin{cases} W^A_{t,k}\, x = \lambda \mathbf{1}, \\ x \in \{0,1\}^l. \end{cases}$$

As previously mentioned, the formulation $(\mathrm{DP_A})$ was used for finding t-designs in [54] [55] [58] [59] [69] [90].

The following proposition shows similar formulations for packings and coverings.

Proposition 2.1.3 *Let A be a permutation group, $l = \left|\binom{V}{k}/A\right|$ and $c_i = |A(K_i)|$, for $i = 1, \ldots, l$. The simple t–(v, k, λ) packing designs admitting automorphism group A are the solutions $x \in \mathbb{R}^l$ of*

$$(\mathrm{PDP_A}) \quad \begin{cases} \text{maximize} & [c_1, \ldots, c_l]\ x \\ \text{subject to} & W_{t,k}^A\ x \le \lambda \mathbf{1}, \\ & x \in \{0,1\}^l. \end{cases}$$

Analogously, the simple t–(v, k, λ) covering designs admitting automorphism group A are the solutions $x \in \mathbb{R}^l$ of

$$(\mathrm{CDP_A}) \quad \begin{cases} \text{minimize} & [c_1, \ldots, c_l]\ x \\ \text{subject to} & W_{t,k}^A\ x \ge \lambda \mathbf{1}, \\ & x \in \{0,1\}^l. \end{cases}$$

For designs with $\lambda = 1$, problems ($\mathrm{DP_A}$) and ($\mathrm{PDP_A}$) can be brought to the form of set partitioning and set packing problems, respectively. Note that problems ($\mathrm{DP_A}$) and ($\mathrm{PDP_A}$) differ from set partitioning and set packing in that $W_{t,k}^A$ is not necessarily a 0-1 matrix. However, any column j containing an element greater than or equal to 2 can be eliminated, since for $\lambda = 1$ any solution x of ($\mathrm{DP_A}$) or ($\mathrm{PDP_A}$) must have $x_j = 0$. Therefore, after performing these eliminations, problems ($\mathrm{DP_A}$) and ($\mathrm{PDP_A}$) with $\lambda = 1$ become special cases of set partitioning and set packing, respectively.

2.1.2 Other integer programming formulations

In this section, we discuss alternative integer programming formulations for BIBDs, i.e. 2-(v, k, λ) designs, based on a quadratic system of equations. Although this formulation is not used in the rest of the thesis, we include it here for completeness. This can be generalized for t-designs with $t > 2$ at the cost of having a system of equations of degree t rather than a quadratic one.

Let X denote the *point-block incidence matrix* of a 2–(v, k, λ) design with b-blocks, that is a $v \times b$ matrix $X = (x_{p,B})$ indexed by the points and blocks of the design and such that

$$x_{p,B} = \begin{cases} 1 & \text{if } p \in B, \\ 0 & \text{otherwise.} \end{cases}$$

A well known property of these matrices is given in the following theorem.

Theorem 2.1.4 *Let I_n and $J_{n\times m}$ denote the $n \times n$ identity matrix and the $n \times m$ matrix of all ones, respectively. A $v \times b$ 0-1 matrix X satisfies*

$$\begin{aligned} XX^T &= (r-\lambda)I_v + \lambda J_{v\times v}, \\ J_{v\times v}X &= kJ_{v\times b}, \end{aligned}$$

if and only if X is the point-block incidence matrix of a 2-(v,k,λ) design with b blocks and point replication equal to r.

A proof of this theorem can be found for example in [98]. Rewriting the equations given by the theorem, we get

$$\text{(QDP)} \quad \begin{cases} \sum_{l=1}^{b} x_{il} = r, & 1 \le i \le v, \\ \sum_{i=1}^{v} x_{il} = k, & 1 \le l \le b, \\ \sum_{l=1}^{b} x_{il}x_{jl} = \lambda, & 1 \le i < j \le v, \\ x_{il} \in \{0,1\} & 1 \le i \le v,\ 1 \le l \le b. \end{cases}$$

In fact, the first (or second) types of inequalities can be removed from the system, since integer solutions of the remaining system will satisfy the first (second) type of inequalities for any admissible parameters.

Wengrzik [103] studied in her Master's thesis an equivalent linear system obtained by introducing extra variables $y_{il}^l = x_{il}x_{jl}$ and given by

$$\text{(LDP)} \quad \begin{cases} \sum_{i=1}^{v} x_{il} = k, & 1 \le l \le b, \\ \sum_{l=1}^{b} y_{ij}^l = \lambda, & 1 \le i < j \le v, \\ y_{ij}^l - x_{il} \le 0, & 1 \le i < j \le v,\ 1 \le l \le b, \\ y_{ij}^l - x_{jl} \le 0, & 1 \le i < j \le v,\ 1 \le l \le b, \\ x_{il} + x_{jl} - y_{ij}^l \le 1, & 1 \le i < j \le v,\ 1 \le l \le b, \\ x_{il} \in \{0,1\} & 1 \le i \le v,\ 1 \le l \le b, \\ y_{ij}^l \in \{0,1\} & 1 \le i < j \le v,\ 1 \le l \le b. \end{cases}$$

Wengrzik [103] studied the polytope associated with (LDP) and proposed a polyhedra-based algorithm for solving such problems. Several instances were solved, although no new designs were produced. Although a thorough comparison was not done, it seems to us that this formulation may be preferable than the formulation in (DP) for designs with $t = 2$ and larger k or λ. For smaller λ or k as well as for larger t, the formulation (DP) seems more suitable. It would be interesting to investigate the relationship between the formulations in (DP) and (LDP), and whether facets found from one model can be translated into facets for the other one.

Another approach for solving these systems was proposed by Mathon [70], by considering the relaxation of (QDP) given by

$$
(\mathrm{QDP}^{\mathrm{R}}) \quad \left\{ \begin{array}{rll}
\sum_{l=1}^{b} x_{il} = r, & 1 \le i \le v, & (2.1) \\
\sum_{i=1}^{v} x_{il} = k, & 1 \le l \le b, & (2.2) \\
\sum_{l=1}^{b} x_{il} x_{jl} = \lambda, & 1 \le i < j \le v, & (2.3) \\
0 \le x_{il} \le 1, & 1 \le i \le v, \ 1 \le l \le b. & (2.4)
\end{array} \right.
$$

Mathon employed a conjugate-gradient method to find local minima for the least-square function associated with the above system, which was iterated by choosing random initial points until a design was found. He empirically observed that whenever a solution of ($\mathrm{QDP}^{\mathrm{R}}$) was found, the integrality condition was satisfied. The following theorem guarantees that this is always the case, and that the relaxed system ($\mathrm{QDP}^{\mathrm{R}}$) is equivalent to the integral one (QDP).

Theorem 2.1.5 *Let v, k, r, λ, b be admissible parameters for a 2–(v, k, λ) design with b blocks and point replication r. Let x be a solution to the system of equations* ($\mathrm{QDP}^{\mathrm{R}}$)*. Then, we have $x_{il} = 0$ or $x_{il} = 1$, for all $1 \le i \le v, 1 \le l \le b$.*

Proof. First, we show that $\sum_{l=1}^{b} x_{il}^2 = r$, for $1 \le i \le v$. Adding the equations (2.3) involving a fixed i, we get $\sum_{j=1, j \ne i}^{v} (\sum_{l=1}^{b} x_{il} x_{jl}) = (v-1)\lambda$. Interchanging the summations, rearranging the left-hand side and using the fact that admissible parameters satisfy $(v -$

$1)\lambda = r(k-1)$, we get $\sum_{l=1}^{b} x_{il} \sum_{j=1,j\neq i}^{v} x_{jl} = r(k-1)$, for all $1 \leq i \leq v$. By (2.2), it follows that $\sum_{j=1,j\neq i}^{v} x_{jl} = k - x_{il}$. Substituting this in the previous equation, we obtain $\sum_{l=1}^{b} x_{il}(k - x_{il}) = r(k-1)$. Therefore, $k\sum_{l=1}^{b} x_{il} - \sum_{l=1}^{b} x_{il}^2 = r(k-1)$, and by (2.1) it follows that $kr - \sum_{l=1}^{b} x_{il}^2 = r(k-1)$. Finally, $\sum_{l=1}^{b} x_{il}^2 = r$ for all $1 \leq i \leq v$.

Consequently, for all $0 \leq i \leq v$, $\sum_{l=1}^{b} x_{il} = \sum_{l=1}^{b}(x_{il})^2$. Hence, $\sum_{l=1}^{b} x_{il}(1-x_{il}) = 0$ and since $0 \leq x_{il} \leq 1$, we have $x_{il}(1-x_{il}) = 0$ for all $1 \leq i \leq v$, $1 \leq l \leq b$. This implies that either $x_{il} = 0$ or $x_{il} = 1$, for all $1 \leq i \leq v, 1 \leq l \leq b$. $\square$

2.2 General set partitioning, set packing and set covering polytopes

In this section, we summarize results on polytopes for more general problems with similar structures to the design problems.

Let A be an $m \times n$ 0-1 matrix and $c \in \mathbb{R}^n$. The set packing, set partitioning and set covering problems are, respectively, given by

$$(\mathrm{PK}) \quad \begin{cases} \max\ c^T x \\ A\ x \leq \mathbf{1}, \\ x \in \{0,1\}^n \end{cases} \qquad (\mathrm{PT}) \quad \begin{cases} \max\ c^T x \\ A\ x = \mathbf{1}, \\ x \in \{0,1\}^n \end{cases} \qquad (\mathrm{CV}) \quad \begin{cases} \min\ c^T x \\ A\ x \geq \mathbf{1}, \\ x \in \{0,1\}^n \end{cases}$$

Let p be an integer number, and consider the generalizations of the previous problems by allowing a constant right-hand side

$$(\mathrm{PK^G}) \quad \begin{cases} \max\ c^T x \\ A\ x \leq p\mathbf{1}, \\ x \in \{0,1\}^n \end{cases} \qquad (\mathrm{PT^G}) \quad \begin{cases} \max\ c^T x \\ A\ x = p\mathbf{1}, \\ x \in \{0,1\}^n \end{cases} \qquad (\mathrm{CV^G}) \quad \begin{cases} \min\ c^T x \\ A\ x \geq p\mathbf{1}, \\ x \in \{0,1\}^n \end{cases}$$

The problem for which the most is known is (PK), due to its equivalence to maximum independent sets of nodes in a graph, as we will see in Section 2.2.1. Its generalization ($\mathrm{PK^G}$) is in the form of maximum independent sets in an independence system, which will be discussed in Section 2.2.2. Both covering problems (CV) and ($\mathrm{CV^G}$) can be brought to the form of maximum independent sets in an independence system by complementing variables, as we discuss in Section 2.2.3. Less is known about the partitioning problems (PT) and (PT^G) and their polytopes, but they can be transformed into set packing or set covering after some adjustment of the objective function (see [9]). Moreover, in the case of the design problems, as we saw in Proposition 2.1.1, the equivalence is trivial, so we can concentrate on packing and covering problems.

All six problems were shown to be NP-hard for general matrix A (see [65]). However, for the special case of design problems, it is not known whether or not the problems lie in P. For surveys on these problems see [9] [13, Part I] [17].

2.2.1 Independent sets and the set packing polytope

The set packing polytope, the one associated with (PK), is given by

$$P(A) = conv\{x \in \{0,1\}^n : Ax \leq \mathbf{1}\}.$$

Let $IS(G)$ be the polytope of the independent sets of nodes for a graph G, that is

$$IS(G) = conv\{x \in \{0,1\}^n : x \text{ is an incidence vector of an independent set of } G\}.$$

The intersection graph of a matrix $A = (a_{ij})$, denoted by G_A, is the graph whose vertices correspond to the columns of A and an edge connects two columns j_1,j_2 if and only if there exists a row i with $a_{ij_1} = a_{ij_2} = 1$.

It is well known that $P(A) = IS(G_A)$. In addition, for any graph G, the edge-node incidence matrix A_G satisfies $IS(G) = P(A_G)$. So, the set packing polytope for a matrix is the independent set polytope for some graph. For this reason, we will use $P(G)$ to denote the polytope $IS(G)$.

A great deal of research has been done on the independent set polytope in the past three decades. We now present the main results regarding this polytope.

Proposition 2.2.1 *(Basic properties) Let A be a 0-1 matrix and let $P(A)$ be the set packing polytope associated with A. Then*

1. *the polytope $P(A)$ is full dimensional;*

2. *the inequalities $x_i \geq 0$ are facet inducing for $P(A)$;*

3. the polytope $P(A)$ is down-monotone, that is, if $x \in P(A)$ then $y \in P(A)$ for all $\mathbf{0} \le y \le x$; in addition, any facet inducing inequality $\alpha^T x \le \alpha_0$, except for $x_i \ge 0$, has all nonnegative coefficients.

The following theorem shows that if G' is an induced subgraph of G, then the facet inducing inequalities for $P(G')$ can be "lifted" to facet inducing inequalities for $P(G)$.

Theorem 2.2.2 *(Lifting of facets for subgraphs (Padberg [82], for odd-holes; Nemhauser and Trotter [79], for general subgraphs)*
Let $G = (V, E)$ be a graph. Let $V' \subseteq V$ and $G' = (V', E')$ be the subgraph of G induced by V'. If $\sum_{j \in V'} \alpha_j x_j \le s$ is a facet inducing inequality for $P(G')$, then there exist integers β_j, $0 \le \beta_j \le s$ such that

$$\sum_{j \in V'} \alpha_j x_j + \sum_{j \in V \setminus V'} \beta_j x_j \le s$$

is a facet inducing inequality for $P(G)$.

In the following, we list some classical classes of valid and facet inducing inequalities for the independent set polytope.

Clique inequalities (Fulkerson [40], Padberg [82]).

A clique of $G = (V, E)$ is a subset of V that induces a complete subgraph of G. A clique is maximal if it is not properly contained in any other clique. Let C be a clique of G. Then, the inequality

$$\sum_{j \in C} x_j \le 1$$

is valid for $P(G)$. Moreover, the above inequality induces a facet of $P(G)$ if and only if C is maximal (with respect to set inclusion).

Odd hole (Padberg [82]) and odd anti-hole (Nemhauser and Trotter[79]) inequalities.

An *odd hole* is a circuit of odd length greater than 3 without chords. An *odd anti-hole* is a graph whose complement is an odd hole.

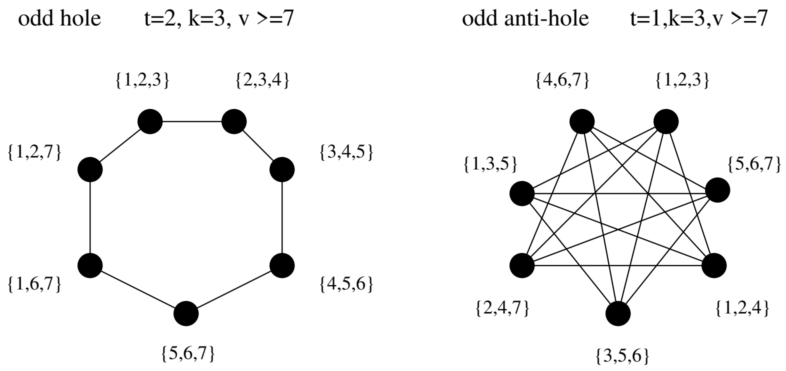

Figure 2.1: Examples of subgraphs of the intersection graph of $W^v_{t,k}$.

If $H \subseteq V$ induces an odd hole G' of G then

$$\sum_{j \in H} x_j \leq \frac{|H| - 1}{2}$$

is a facet inducing inequality of $P(G')$.

If $A \subseteq V$ induces an odd anti-hole G' of G then

$$\sum_{j \in A} x_i \leq 2$$

is a facet inducing inequality of $P(G')$.

Figure 2.1 shows an odd hole and an odd anti-hole for graphs associated with packing design problems.

Web and anti-web inequalities (Trotter [100]).

Webs are generalizations of cliques, holes and anti-holes. Let n and k be integers with $n \geq 2$ and $1 \leq k \leq \frac{n}{2}$. A *web* $W(n,k)$ is the graph with vertices $V_n = \{1, \ldots, n\}$ and edges defined by: $\{i,j\} \in E$ if and only if $(j-i) \bmod n \in \{k, k+1, ..., n-k\}$. Observe that $W(n,1)$ is a clique, $W(2s+1,s)$ is an odd hole and $W(2s+1,2)$ is an odd anti-hole. The inequality

$$\sum_{j=1}^{n} x_j \leq k$$

is facet inducing for $P(W(n,k))$ if and only if n and k are relatively prime.

An *anti-web* is a complement of a web, denoted by $\overline{W}(n,k)$. The inequality

$$\sum_{j=1}^{n} x_j \leq \lfloor n/k \rfloor$$

is facet inducing for $P(\overline{W}(n,k))$ if and only if n and k are relatively prime.

Wheel inequalities (see [68])

An odd wheel of center v_0 is a graph with vertices $\{v_0, v_1, \ldots, v_n\}$ such that $v_1, \ldots, v_n$ forms an odd cycle and v_0 is connected by an edge to all of the other vertices. If $W \subseteq V$ induces an odd wheel G' of G with center v_o then

$$\sum_{i \in W \setminus \{v_0\}} x_j + \frac{|W|-2}{2} x_{v_0} \leq \frac{|W|-2}{2}$$

is a facet inducing inequality for $P(G')$.

Generalizations of these simple wheel inequalities by subdividing edges are proposed by Barahona and Mahjoub [11] and Cheng and Cunningham [19].

Rank inequalities

Let $G = (V,E)$ be a graph and denote by $\alpha(G)$ the independence (or stability) number of G, i.e. the size of a maximum independent set of G. The inequality

$$\sum_{i \in V} x_i \leq \alpha(G)$$

is clearly valid for $P(G)$. The following theorems give sufficient or necessary conditions for this type of inequality to define a facet for $P(G)$.

An edge e of a graph G is said to be *critical* if $\alpha(G-e) = \alpha(G)+1$.

Theorem 2.2.3 *(A sufficient condition for facetness - Chvátal [22])*
Let $G = (V,E)$ be a graph and let E^* be a set of critical edges. If $G^* = (V,E^*)$ is connected then the inequality

$$\sum_{i \in V} x_i \leq \alpha(G)$$

defines a facet of $P(G)$.

Given any proper subset V_1 of V, the *cutset* $C = (V_1, V \setminus V_1)$ is the (possibly empty) set of edges joining the nodes in V_1 to those in $V \setminus V_1$. Let G_1 and G_2 be the subgraphs induced by V_1 and $V \setminus V_1$, respectively. The cutset $C = (V_1, V \setminus V_1)$ is called *α-critical*, if $\alpha(G_1) + \alpha(G_2) \geq \alpha(G) + 1$.

Theorem 2.2.4 *(A necessary condition for facetness - Balas and Zemel [10])*
If the inequality

$$\sum_{i \in V} x_i \leq \alpha(G)$$

defines a facet of $P(G)$, then every cutset of G is α-critical.

Theorem 2.2.5 *(A weaker sufficient condition for facetness - Balas and Zemel [10])*
Let $G = (V, E)$ be a graph with E^* its set of critical edges and such that either $G^* = (V, E^*)$ is connected, or G has an α-critical cutset $C = (V_1, V_2 = V \setminus V_1)$ satisfying the following conditions (for G_i the subgraph of G induced by V_i, $i = 1, 2$):

1. *the inequality $\sum_{j \in V_i} x_j \leq \alpha(G_i)$ defines a facet of $P(G_i)$;*

2. *every maximum-cardinality independent set of G_i is contained in some maximum-cardinality independent node set of G.*

Then, $\sum_{i \in V} x_i \leq \alpha(G)$ defines a facet of $P(G)$.

Theorem 2.2.6 *(Necessary and sufficient condition for extension of rank facets - Balas and Zemel [10])*
Let $G^1 = (V^1, E^1)$ be the subgraph of $G = (V, E)$ induced by $V^1 \subsetneq V$, and such that

$$\sum_{i \in V^1} x_i \leq \alpha(G^1) \tag{2.5}$$

defines a facet of $P(G^1)$. Then (2.5) defines a facet of $P(G)$ if and only if for every $k \in V \setminus V^1$, the cutset $(\{k\}, V^1)$ of $G[V^1 \cup \{k\}]$ (the subgraph of G induced by $V^1 \cup \{k\}$) is not α-critical.

Other properties and inequalities

Proposition 2.2.7 *(2-connectedness of facet support - see [13])*
Let G be a graph. If $\alpha^T x \leq \alpha_0$ defines a facet of $P(G)$, then its support $G[\text{supp } \alpha^T]$ is 2-(node)-connected.

Inequalities derived from quadratic relaxations of the set packing polytope can be found in [68], and from other relaxations in [13].

Adjacency in the set packing polytope

Two vertices are adjacent in a polytope if they lie in a face of dimension one (also called an edge) of the polytope.

Theorem 2.2.8 *(Necessary and sufficient condition for adjacency - Chvátal [22])*
Let $G = (V, E)$ be a graph, let V_1 and V_2 be independent sets of G and x^1,x^2 their respective incidence vectors. Then x^1 and x^2 are adjacent in $P(G)$ if and only if the subgraph of G induced by $(V_1 \setminus V_2) \cup (V_2 \setminus V_1)$ is connected.

Other adjacency results, especially for partitioning problems, can be found in [8] [9].

2.2.2 Independence systems and the generalized set packing polytope

An independence system is a pair $(I, \mathcal{I})$ where I is an n-set and $\mathcal{I}$ is a family of subsets of I, with the property that $I_1 \subseteq I_2 \in \mathcal{I}$ implies $I_1 \in \mathcal{I}$; the individual members of $\mathcal{I}$ are called independent sets. Sets $J \subseteq I$ such that $J \notin \mathcal{I}$ are said to be *dependent* and minimal such sets are called *circuits*. An independence system is characterized by its family of circuits.

Examples:

1. Let $G = (V, E)$ be a graph and $\mathcal{I}_G$ be its family of independent sets. Then $(V, \mathcal{I}_G)$ form an independence system with E being its family of circuits.

2. Let $I = [1, n]$ and let A be a 0-1 matrix and p a positive integer. Let $\mathcal{I}_A$ be the family of all the 0-1 solutions x to $Ax \leq p\mathbf{1}$. Then $(I, \mathcal{I}_A)$ is an independence system whose family of circuits is given by

$$E_A = \{C \subseteq I : |C| = p+1 \text{ and there exists a row } i \text{ of } A \text{ with } a_{ij} = 1 \text{ for all } j \in C\}. \quad (2.6)$$

Observe that in the two previous examples the independence system is r-regular, i.e. all its circuits have same cardinality r ($r = 2$ and $r = p + 1$ for the two examples, respectively). We will focus on r-regular independence systems, but we remark that an independence system is not so in general.

Let $P_p(A)$ be the polytope corresponding to ($\mathrm{PK_G}$), i.e.

$$P_p(A) = conv\{x \in \{0, 1\}^n : Ax \leq p\mathbf{1}\}.$$

For $P_p(A)$, the *intersection hypergraph* of A is given by $H_A = (I, E_A)$, where E_A is given in (2.6). Thus, similarly to ordinary set packings, the generalized set packings correspond to regular independence systems. We will denote by $P(S)$ the polytope of the independent sets of the independence system S.

Clique inequalities for regular independence systems

Let $(I, \mathcal{I})$ be a p-regular independence system. A subset $I' \subseteq I$ is said to be a clique if $|I'| \geq p$ and all p-subsets of I' are circuits of $(I, \mathcal{I})$.

Theorem 2.2.9 *(Nemhauser and Trotter [79])*
Suppose $I' \subseteq I$ is a maximal (with respect to set-inclusion) clique in a p-regular independence system $S = (I, \mathcal{I})$. Then

$$\sum_{i \in I'} x_i \leq p - 1$$

defines a facet of $P(S)$.

Cliques for more general independence systems, as well as generalized odd-holes and anti-holes can be found in [35] [92].

Lifting for independence systems

A lifting theorem, similar to the one for independent sets in graphs, generalizes to independence systems.

Theorem 2.2.10 *(Nemhauser and Trotter [79])*
Let $S' = (I', \mathcal{I}') \subseteq S = (I, \mathcal{I})$. If

$$\sum_{i \in I'} \alpha_j x_j \leq s$$

is a facet inducing inequality for $P(S')$, then there exist integers β_j, $0 \leq \beta_j \leq s$ such that

$$\sum_{j \in I'} \alpha_j x_j + \sum_{j \in I \setminus I'} \beta_j x_j \leq s$$

is a facet inducing inequality for $P(S)$.

2.2.3 The set covering polytope

The set covering polytope is the polytope corresponding to (CV) given by

$$C(A) = conv\{x \in \{0,1\}^n : Ax \geq \mathbf{1}\}.$$

Similarly, the generalized set covering problem ($\mathrm{CV^G}$) has its polytope given by

$$C_p(A) = conv\{x \in \{0,1\}^n : Ax \geq p\mathbf{1}\}.$$

The set covering polytope is known to be equivalent to the independence system polytope through complementing variables [62] [81]. Indeed, the same is true for the generalized set covering polytope. Applying the transformation $y := \mathbf{1} - x$, we obtain the two equivalent problems:

$$(\mathrm{CP_p}) \quad \begin{cases} \min \ c^T(\mathbf{1} - y) \\ A\ (\mathbf{1} - y) \geq p\mathbf{1}, \\ (\mathbf{1} - y) \in \{0,1\}^n \end{cases} \qquad (\mathrm{IP}) \quad \begin{cases} c^T\mathbf{1} - \max \ c^T y \\ A\ y \leq A\mathbf{1} - p\mathbf{1}, \\ y \in \{0,1\}^n \end{cases}$$

Moreover, denoting by $I_p(A)$ the polytope associated with (IP), the facets of the two polytopes are related as follows.

Corollary 2.2.11 *The inequality $\alpha^T x \geq \alpha_0$ induces a facet for $C_p(A)$ if and only if the inequality $\alpha^T x \leq \alpha^T \mathbf{1} - \alpha_0$ induces a facet for $I_p(A)$.*

In addition, if A has a constant row sum s, then $I_p(A)$ corresponds to the polytope for uniform independence systems $P_r(A)$, for $r = s - p$. The matrix W_{tk}^v of the design problem satisfies this property.

The study of the set covering polytope is way behind the one for the set packing polytope. The equivalence of set coverings and independence systems may explain this unbalance, since studying independent sets in hypergraphs is more complex than in graphs. The concept of intersecting graphs for the set packing matrix has a parallel with the one for matrix minors of the set covering matrix, but the latter presents a "lack of locality" in the sense that inequalities valid for the set covering associated with a matrix minor are not even guaranteed to be valid for the original polytope, and lifting is more involved. Nevertheless, some research has been done on the set covering polytope, which we briefly discuss.

Generalized anti-web inequalities are studied in [62] [89], generalized web inequalities in [89] and composition of rank facets in [81]. The inequalities with coefficients in $\{0, 1, 2\}$, are characterized in [7]. Other inequalities can be found in [28]. Conditions for rank facets of the set covering polytope to be facet defining are given in [13] [62] . In addition, as seen in Corollary 2.2.11, the references for independence systems from section 2.2.2 are relevant here.

2.3 Design polytopes

In this section, we show new results about the design polytopes $T_{t,v,k,\lambda}$, $P_{t,v,k,\lambda}$ and $C_{t,v,k,\lambda}$, which are the main contributions of this chapter.

Section 2.3.1 contains basic properties for the three polytopes. In Section 2.3.2, we examine the polytope $P_{t,v,k,\lambda}$. Two classes of inequalities are studied, namely the generalized clique and subpacking inequalities. A characterization of generalized cliques in terms of intersecting set systems is given (Proposition 2.3.6 and Corollary 2.3.7), as well as connections to results to be proved in Chapter 3. We derive a new class of inequalities which we call subpacking inequalities (Theorem 2.3.9) and we give conditions under which they induce facets (Proposition 2.3.11 to Proposition 2.3.14, and Table 2.1). Separation results for both classes of inequalities are presented (Corollary 2.3.8 and Corollary 2.3.10). In Section 2.3.3, the polytopes for anti-Pasch and m-sparse triple systems are studied and anti-Pasch and m-sparse facets are given (Theorem 2.3.16 and Theorem 2.3.20), as well as separation results (Corollary 2.3.21). In Section 2.3.4, connections between adjacency on $T_{t,v,k,\lambda}$ and null-designs are explored (Theorem 2.3.22).

2.3.1 Basic properties

In this section, we prove some results regarding general t-designs, packings and coverings.

Let $e^i \in \mathbb{R}^{\binom{v}{k}}$, $1 \leq i \leq \binom{v}{k}$, be defined by $(e^i)_i = 1$ and $(e^i)_j = 0$ for $j \neq i$. The following theorem gives information on the dimension of the design polytopes.

Proposition 2.3.1 *(Dimension)*

i. If $k \leq v - t$ then $dim T_{t,v,k,\lambda} \leq \binom{v}{k} - \binom{v}{t}$.

ii. $P_{t,v,k,\lambda}$ is full dimensional.

iii. If $\lambda < \binom{v-t}{k-t}$ then $C_{t,v,k,\lambda}$ is full dimensional.

Proof.

i. This is a direct consequence of the fact that $rank(W^v_{t,k}) = \binom{v}{t}$, when $k \leq v - t$ (see [44] for a proof). Since the inequalities in $W^v_{t,k} x \leq \lambda \mathbf{1}$ are part of the equality representation of $T_{t,v,k,\lambda}$, say $(A^=, b^=)$, it follows that $rank(A^=, b^=) \geq rank(W^v_{t,k}, \lambda\mathbf{1}) = \binom{v}{t}$. Therefore, by Proposition 1.2.2, $dim T_{t,v,k,\lambda} \leq \binom{v}{k} - \binom{v}{t}$.

ii. Since $\mathbf{0}, e^1, \ldots, e^{\binom{v}{k}} \in P_{t,v,k,\lambda}$ are affine independent, the result follows.

iii. Since $\lambda < \binom{v-t}{k-t}$, we see that $\mathbf{1} - e^1, \ldots, \mathbf{1} - e^{\binom{v}{k}}, \mathbf{1} \in C_{t,v,k,\lambda}$, which are affine independent. □

Proposition 2.3.2 *(Non-negativity constraints)*
The inequalities $x_i \geq 0$, $i = 1, \ldots, \binom{v}{k}$, *are facet inducing for* $P_{t,v,k,\lambda}$. *Moreover, if* $\lambda \leq \binom{v-t}{k-t} - 1$, *then they are facet inducing for* $C_{t,v,k,\lambda}$.

Proof. It is enough to show $\binom{v}{k}$ affine independent points in each polytope that satisfy $x_i = 0$. For $P_{t,v,k,\lambda}$, the points in $\{e_j\}_{j \neq i} \cup \{\mathbf{0}\}$ satisfy this property. If $\lambda \leq \binom{v-t}{k-t} - 1$, then $\{\mathbf{1} \setminus \{e_i, e_j\}\}_{j \neq i} \cup \{\mathbf{1} \setminus e_i\} \subseteq C_{t,v,k,\lambda}$. The result follows from the affine independence of these points. □

Proposition 2.3.3 *(Relation between packing and covering numbers)*
Let $v > k > t$, λ *be integer numbers and* $\lambda' = \binom{v-t}{k-t} - \lambda$, $\lambda'' = \lambda\binom{v-t}{k-t} - \lambda$. *Then,*

$$\begin{aligned} \overline{C}_\lambda(v,k,t) &= \binom{v}{k} - \overline{D}_{\lambda'}(v,k,t), \\ C_\lambda(v,k,t) &= \lambda\binom{v}{k} - D_{\lambda''}(v,k,t). \end{aligned}$$

Moreover, $y \in \{0,1\}^n$ *is the incidence vector of a* t–(v,k,λ) *covering (simple* t–(v,k,λ) *covering) if and only if* $\mathbf{1} - y$ *is the incidence vector of a* t–(v,k,λ'') *packing (simple* t–(v,k,λ') *packing).*

Proof. This is a direct consequence of the equivalence between the generalized set covering problem ($\mathrm{CP_p}$) and the independence system problem (IP) seen in Section 2.2.3. For simple designs take $c = \mathbf{1}$, $A = W^v_{t,k}$ and $p = \lambda$ in ($\mathrm{CP_p}$) and (IP), and observe that the optimum values for these problems are equal to $\overline{C}_\lambda(v,k,t)$ and $\overline{D}_{\lambda'}(v,k,t)$, respectively. For designs with possibly repeated blocks, assume A to have the same columns as $W^v_{t,k}$ repeated λ times. This matrix does define the correct problem for designs with repeated blocks, since any t–(v,k,λ) packing and any minimum t–(v,k,λ) covering cannot have a block repeated more than λ times. After some simple algebraic manipulations, the proof is complete. □

2.3.2 Inequalities for the polytope $P_{t,v,k,\lambda}$

We concentrate on the packing design polytope $P_{t,v,k,\lambda}$. As we have seen in previous sections, $P_{t,v,k,\lambda} = P_\lambda(W^v_{t,k})$, i.e. the polytope associated with the generalized set packing problem whose matrix is $W^v_{t,k}$ and right-hand side is λ.

$$
W^5_{2,3} = \begin{array}{cc}
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1
\end{bmatrix}
&
\begin{array}{c}
\{1,2\} \\ \{1,3\} \\ \{1,4\} \\ \{1,5\} \\ \{2,3\} \\ \{2,4\} \\ \{2,5\} \\ \{3,4\} \\ \{3,5\} \\ \{4,5\}
\end{array}
\\
\begin{array}{cccccccccc}
\{1,2,3\} & \{1,2,4\} & \{1,2,5\} & \{1,3,4\} & \{1,3,5\} & \{1,4,5\} & \{2,3,4\} & \{2,3,5\} & \{2,4,5\} & \{3,4,5\}
\end{array}
&
\end{array}
$$

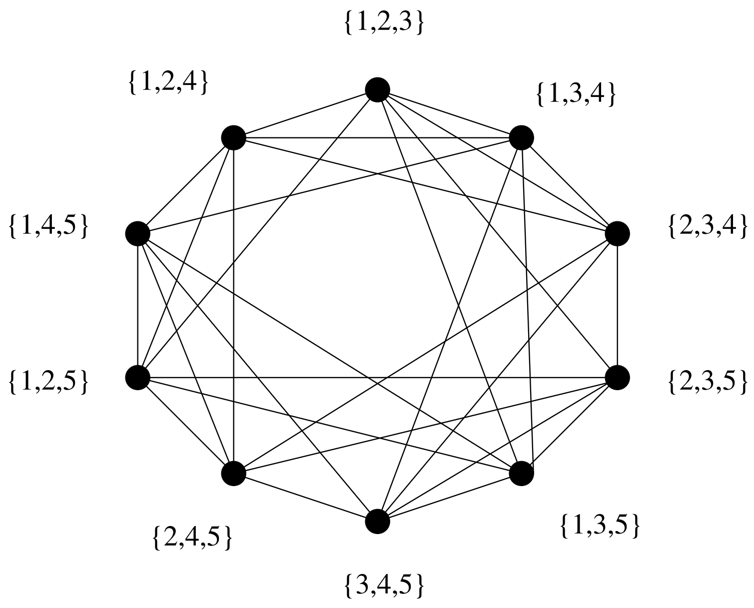

Figure 2.2: Intersection graph of $W^5_{2,3}$.

For $\lambda = 1$, the problem is an ordinary set packing. The intersection graph $G^v_{t,k}$ of the matrix $W^v_{t,k}$ is the graph whose vertices correspond to the k-subsets of $[1, v]$ and two k-subsets are connected by an edge if and only if they intersect in at least t points. Figure 2.2 shows an example of such a graph.

For $\lambda > 1$, the problem is the same as finding a maximum independent set in an independence system. The circuits of this independence system are the sets of $(\lambda + 1)$ k-subsets of $[1, v]$ sharing a common t-subset, as stated by the following proposition.

Proposition 2.3.4 *(Packings and independence systems)*
A t–(v, k, λ) packing is an independent set of the independence system given by the circuits

$$\mathcal{C} = \left\{ \{K_1, K_2, \ldots, K_{\lambda+1}\} : K_i \in \binom{[1, v]}{k} \text{ and } |K_1 \cap K_2 \cap \ldots \cap K_{\lambda+1}| \geq t\} \right\}. \quad (2.7)$$

A clique in such a $(\lambda + 1)$-regular independence system corresponds to a group of k-subsets of $[1, v]$ with the following intersecting properties.

Definition 2.3.5 *(s-wise t-intersecting set systems)*
Given $s \geq 2$ and $v, t \geq 1$, a family $\mathcal{A}$ of subsets of $[1, v]$ is said to be s-wise t-intersecting, *if any s members $A_1, \ldots, A_s$ of $\mathcal{A}$ are such that $|A_1 \cap \ldots \cap A_s| \geq t$. A family $\mathcal{A}$ is said to be* k-uniform *if every member of $\mathcal{A}$ has cardinality k. Let $I^s(v, k, t)$ denote the set of all k-uniform s-wise t-intersecting families of subsets of $[1, v]$. Let $MI^s(v, k, t)$ denote the set of all families in $I^s(v, k, t)$ that are maximal with respect to set inclusion (i.e. $\mathcal{A} \in I^s(v, k, t)$ such that for any $\mathcal{B} \in I^s(v, k, t)$, if $\mathcal{B} \supseteq \mathcal{A}$ then $\mathcal{B} = \mathcal{A}$).*

Proposition 2.3.6 *(Characterization of generalized cliques)*
Let $\mathcal{A}$ be a family of k-subsets of $[1, v]$. Then $\mathcal{A}$ is a generalized clique for the independence system associated with a t–(v, k, λ) packing if and only if $\mathcal{A} \in I^{(\lambda+1)}(v, k, t)$. Moreover, a clique $\mathcal{A}$ is maximal if and only if $\mathcal{A} \in MI^{(\lambda+1)}(v, k, t)$.

Proof. By the definition of clique in an independence system (see Section 2.2.2), $\mathcal{A}$ corresponds to a clique for the set packing independence system if and only if $\mathcal{A}$ is a family of k-subsets of $[1, v]$ such that all subfamilies of $\mathcal{A}$ with $(\lambda + 1)$ elements are circuits. By

Proposition 2.3.4, this is equivalent to $\mathcal{A} \in I^{(\lambda+1)}(v,k,t)$. Clearly, the clique is maximal if and only if $\mathcal{A} \in MI^{(\lambda+1)}(v,k,t)$. □

The following corollary characterizes the clique inequalities for the polytope $P_{t,v,k,\lambda}$.

Corollary 2.3.7 *(Generalized clique inequalities)*
Let $\mathcal{A} \in I^{(\lambda+1)}(v,k,t)$. Then, the inequality

$$\sum_{K \in \mathcal{A}} x_K \leq \lambda$$

is valid for the polytope $P_{t,v,k,\lambda}$. Moreover, the above inequality induces a facet of $P_{t,v,k,\lambda}$ if and only if $\mathcal{A} \in MI^{(\lambda+1)}(v,k,t)$.

Proof. It follows from Proposition 2.3.6 and the clique inequalities described in page 34.

In light of Proposition 2.3.6, it is natural to ask whether it is possible to determine the form of all clique inequalities for $P_{t,v,k,\lambda}$, leading to the problem discussed in Chapter 3:

> **Problem.** Given s, n, k and t, list all families of $MI^s(v,k,t)$; or alternatively, list all nonisomorphic families of $MI^s(v,k,t)$.

This problem turns out to be, in general, a nontrivial problem in extremal set theory. However, we obtained some results, including the classification of cliques for $k - t \leq 2$ which will be presented in Chapter 3. We will list these results here to summarize their main consequences on clique inequalities; the reader is referred to Chapter 3 for proofs.

Determination of generalized clique facets for some set of packing parameters:

1. For any $t \geq 1$, $v \geq t+3$, $\lambda = 1$ and $k = t+1$, there exists exactly two distinct (up to isomorphism) clique facets for $P_{t,v,t+1,1}$, namely

$$\left\{\begin{aligned} &\sum_{K \in \binom{[1,v]}{t+1} : K \supseteq T} x_K \leq 1, \qquad \text{for all } T \in \binom{[1,v]}{t}, && (2.8)\\ &\sum_{K \in \binom{L}{t+1}} x_K \leq 1, \qquad \text{for all } L \in \binom{[1,v]}{t+2}. && (2.9) \end{aligned}\right.$$

2. For any $t \geq 1$, $v \geq t+6$, $\lambda = 1$ and $k = t+2$, there exists exactly 15 distinct types of clique facets for $t = 1$ and 17 distinct types of clique facets for $t \geq 2$. These cliques are given explicitly for $t = 1$ and $t = 2$, and by a construction for $t \geq 3$ (see Table 3.1 for their forms).

3. For any $t \geq 1$, $v \geq t+2$, $\lambda \geq 2$ and $k = t+1$, there exists exactly one distinct (up to isomorphism) generalized clique, namely:

$$\sum_{K \in \binom{[1,v]}{t+1} : K \supseteq T} x_K \leq 1.$$

4. For arbitrary $k > t$ and λ, there exists a $v_0 = v_0(t, \lambda, k)$ such that all clique facets for $v \geq v_o$ are determined by those for v_0.

The knowledge of the clique structure can help us designing separation algorithms. For example, for the case of $k = t+1$ and $\lambda = 1$ the separation of clique inequalities turns out to be quite simple, as shown in the following algorithm.

Algorithm: Separation of clique inequalities for $P_{t,v,t+1,1}$
Input: a fractional solution $\overline{x}$ to (PDP)
Output: a violated clique inequality or "There are no violated cliques"

let $G^v_{t,t+1} = \left(\binom{[1,v],}{K}, E\right)$ be the intersection graph of $W^v_{t,t+1}$.
for every edge $\{K_1, K_2\} \in E$
  take $L = K_1 \cup K_2$
  if $\sum_{K \in \binom{L}{t+1}} \overline{x}_K > 1$ then
    return "Violated clique: ", L
return "There are no violated cliques."

Remark: In Chapter 4, we will see that we do not need to examine every edge, but just the ones in the "fractional subgraph", i.e., only the edges connecting $\{K_1, K_2\}$, where $0 < x_{K_1}, x_{K_2} < 1$. Moreover, the algorithm can be easily adapted to find all violated cliques, instead of halting whenever the first violated clique is found.

Recall from page 21 our measure of complexity.

Corollary 2.3.8 *(Separation of cliques in $P_{t,v,t+1,1}$)*

The clique facets in $P_{t,v,t+1,1}$ can be separated in polynomial time.

Proof. The statement is implied by the correctness of the previous algorithm. Any fractional solution $\overline{x}$ to (PDP) must satisfy $W^v_{t,t+1}\overline{x} \leq 1$. So, inequalities (2.8) are satisfied and the only possible violated cliques are the ones in (2.9). Note that there is exactly one clique of type (2.9) passing through each edge $\{K_1, K_2\}$ in the graph $G^v_{t,t+1}$, for $|K_1 \cup K_2| = t+2$. This shows the correctness of the algorithm. The polynomiality can be checked by noticing that every iteration takes polynomial number of steps and the number of iterations is at most the square of the number of variables in the problem. □

It would be interesting to generalize the previous algorithm for the case $k = t+2$, in which there are 17 different kinds of cliques.

In the following, we discuss another type of inequality for the packing design polytope. The subpacking inequalities come from the simple fact that in a t–(v, k, λ) packing $(V, \mathcal{B})$ any $\mathcal{B}' \subseteq \mathcal{B}$ with $V' = \cup_{B' \in \mathcal{B}'} B'$ gives rise to a t-$(|V'|, k, \lambda)$ packing $(V', \mathcal{B}')$.

Proposition 2.3.9 *(Subpacking inequality)*

Let $S \subseteq [1, v]$ and $\overline{D}_\lambda(n, k, t)$ be the size of a maximum simple t–(n, k, t) packing. Then, the inequality given by

$$\sum_{K \in \binom{S}{k}} x_K \leq \overline{D}_\lambda(|S|, k, t) \tag{2.10}$$

is valid for $P_{t,v,k,\lambda}$.

As v grows, we do not expect to identify all such inequalities, since they rely on the knowledge of the packing number $\overline{D}_\lambda(|S|, k, t)$, $|S| \leq v$, which is one of the objectives of solving the packing design problem. Nevertheless, it is feasible to determine the packing number for some small values of $|S|$ and then solve the separation problem for subpacking inequalities with small $|S|$, as stated by the following corollary.

Corollary 2.3.10 *(Separation of subpacking inequalities)*

Let C be a constant. Subpacking inequalities with $|S| \leq C$ can be separated in polynomial time.

Proof. For C constant and $s \leq C$, the packing numbers $\overline{D}_\lambda(s,k,t)$ can be determined in polynomial time. Moreover, there is a polynomial number, namely $\sum_{s=k+1}^{C} \binom{v}{s}$, of such inequalities. □

Note that a subpacking inequality is not always facet defining, as illustrated by the following proposition.

Proposition 2.3.11 *If there exists a simple t-(v,k,λ) design, then*

$$\sum_{K \in \binom{[1,v]}{k}} x_K \leq \overline{D}_\lambda(v,k,t) \tag{2.11}$$

does not induce a facet of $P_{t,v,k,\lambda}$.

Proof. Since there exists a simple t-(v,k,λ) design, it follows that $\overline{D}_\lambda(v,k,t) = b = \lambda\binom{v}{t}/\binom{k}{t}$. But then equation (2.11) is obtained by adding the inequalities in $W^v_{t,k}x \leq \lambda\mathbf{1}$, and thus cannot induce a facet. □

A natural problem is to determine, whenever a subpacking inequality does define a facet for $P_{t,\overline{v},k,1}$, whether it also defines a facet for $P_{t,v,k,1}$, for $v \geq \overline{v}$. In the case of $\lambda = 1$, the subpacking inequalities are rank inequalities (see in Section 2.2.1) for independent sets in graph $G^v_{t,k}$. The inequality in (2.10) is the rank inequality associated with the subgraph of $G^v_{t,k}$ induced by S. The following proposition tell us when a subpacking inequality can be extended.

Proposition 2.3.12 *(Extension of subpacking facets)*
Let $\overline{v} > k > t$. Suppose the subpacking inequality

$$\sum_{K \in \binom{[1,\overline{v}]}{k}} x_K \leq \overline{D}_1(\overline{v},k,t) \tag{2.12}$$

defines a facet of $P_{t,\overline{v},k,1}$. Then, (2.12) defines a facet of $P_{t,v,k,1}$, $v > \overline{v}$ if and only if for all $L \in \binom{[1,v]}{k} \setminus \binom{[1,\overline{v}]}{k}$ there exists a t-$(\overline{v},k,1)$ packing design $([1,\overline{v}],\mathcal{B})$ such that $([1,v],\mathcal{B}\cup\{L\})$ is a t-$(v,k,1)$ packing design.

Proof. Let $v > \overline{v}$ and let $L \in \binom{[1,v]}{k} \setminus \binom{[1,\overline{v}]}{k}$. Then, the cutset $\left(\binom{[1,\overline{v}]}{k}, \{L\}\right)$ of the subgraph of $G^v_{t,k}$ induced by $\binom{[1,\overline{v}]}{k} \cup \{L\}$ is not critical if and only if the condition of the proposition is satisfied. The proof is completed by applying Theorem 2.2.6. □

In the following corollary, we show that the extension is always possible when $k = t+1$.

Corollary 2.3.13 *Let $\overline{v} > k > t$ with $k = t+1$. If the inequality*

$$\sum_{K \in \binom{[1,\overline{v}]}{t+1}} x_K \leq \overline{D}_1(\overline{v}, t+1, t) \tag{2.13}$$

defines a facet of $P_{t,\overline{v},t+1,1}$, then it also defines a facet for $P_{t,v,t+1,1}$, for all $v \geq \overline{v}$.

Proof. Let $v > \overline{v}$, $L \in \binom{[1,v]}{t+1} \setminus \binom{[1,\overline{v}]}{t+1}$ and $L^I = L \cap [1, \overline{v}]$. Then $|L^I| \leq t$. Let $\mathcal{P} = ([1,\overline{v}], \mathcal{B})$ be any t-$(\overline{v}, t+1, 1)$ packing design with $|\mathcal{B}| = \overline{D}_1(\overline{v}, t+1, t)$. If the inequality (2.13) defines a facet of $P_{t,\overline{v},t+1,1}$, by Proposition 2.3.11, $\mathcal{P}$ cannot be a t-design. Thus, there exists a t-subset $T \subseteq [1, \overline{v}]$ not covered by $\mathcal{P}$. Let π be any permutation on $[1, \overline{v}]$ such that $\pi(T) \supseteq L^I$. Let $\mathcal{P}' = \pi(\mathcal{P})$, and denote its blocks by $\mathcal{B}'$. Then $([1, v], \mathcal{B}' \cup \{L\})$ is a t–$(v, t+1, 1)$ design with $\overline{D}_1(\overline{v}, t+1, t) + 1$ blocks. Therefore, by Proposition 2.3.12, inequality (2.13) defines a facet of $P_{t,v,t+1,1}$. $\square$

Facet inducing subpacking inequalities

We will analyze for some small $|S|$, $k = 3$ and $t = 2$, whether the corresponding packing inequalities define facets. The results are summarized in Table 2.1 and open cases are indicated by question marks. Some of the table entries come from the following proposition.

Proposition 2.3.14 *(Miscellaneous results for subpacking inequalities)*
The subpacking inequalities

$$\begin{array}{llll} \text{(i)} & \displaystyle\sum_{K \in \binom{S_1}{3}} x_K & \leq 2, & |S_1| = 5, \\ \text{(ii)} & \displaystyle\sum_{K \in \binom{S_2}{3}} x_K & \leq 13, & |S_2| = 10, \end{array}$$

induce facets of $P_{2,v,3,1}$, $v \geq 5$ and $v \geq 10$, respectively. The subpacking inequality

$$\text{(iii)} \qquad \sum_{K \in \binom{S_3}{3}} x_K \leq 4, \quad |S_3| = 6,$$

does not induce a facet of $P_{2,v,3,1}$, for any $v \geq 6$.

$\overline{v}$	$\overline{D}_1(\overline{v},3,2)$	$\sum_{K\in\binom{S}{3}} x_K \le \overline{D}_1(\overline{v},3,2)$, $\lvert S\rvert = \overline{v}$, induce facets of $P_{2,v,3,1}$, $v \ge \overline{v}$	reference
4	1	Yes	maximal clique - Theorem 3.4.1
5	2	Yes	Proposition 2.3.14
6	4	No	Proposition 2.3.14
7	7	No	$\exists$ STS(7) - Proposition 2.3.11
8	8	?	-
9	12	No	$\exists$ STS(9) - Proposition 2.3.11
10	13	Yes	Proposition 2.3.14
11	17	?	-
12	20	?	-
13	26	No	$\exists$ STS(13) - Proposition 2.3.11
14	28	?	-
15	35	No	$\exists$ STS(15) - Proposition 2.3.11

Table 2.1: Facet inducing subpacking inequalities for $k = 3$ and $t = 2$.

Proof. First we prove that inequality (i) induces a facet of $P_{2,v,3,1}$, for $v \ge 5$. In light of Corollary 2.3.13, it is enough to prove it for $v = 5$. By Theorem 2.2.3, it is sufficient to prove that every edge in $G^5_{2,3}$ is critical. Indeed, any edge of $G^5_{2,3}$ connects vertices of the form $\{a,b,c\}, \{a,b,d\}$ for distinct a,b,c,d; thus, removing such an edge, makes the collection $\{\{a,b,c\},\{a,b,d\},\{c,d,e\}\}$ a valid independent set. Hence, any edge in $G^v_{2,3}$ is critical.

Similarly, in order to prove that inequality (ii) induces a facet of $P_{2,v,3,1}$, $v \ge 10$, it is sufficient to show that any edge of $G^{10}_{2,3}$ is critical. Take any maximum 2–$(10,3,1)$ packing, i.e. a packing with 13 blocks. Then, from the $\binom{10}{2} = 45$ possible pairs in $[1,10]$, only $13 \times 3 = 39$ are covered. This leaves 6 uncovered pairs that cannot be all disjoint, since they are taken from 10 points. Therefore, there are two uncovered pairs $\{a,b\}$,$\{a,c\}$ for distinct a,b,c, and obviously $\{b,c\}$ is covered, otherwise $\{a,b,c\}$ would belong to this maximal packing. Let d be the point that appears together with $\{b,c\}$. Removing the edge that connects $\{a,b,c\}$ and $\{b,c,d\}$ increases the size of the maximum packing by 1, and so this edge is critical. Since the same is true for every edge (just apply a permutation on $[1,10]$ to get a suitable isomorphic maximum packing), we conclude that every edge is critical.

To prove that inequality (iii) does not induce a facet of $P_{2,v,3,1}$, $v \geq 6$, assume w.l.o.g. that $S_3 = [1,6]$. Adding the valid inequalities $\sum_{K \in \binom{S}{3}: T \subseteq K} x_K \leq 2$ for all $S \in \binom{[1,6]}{5}$, we get $3 \sum_{K \in \binom{[1,6]}{3}} x_K \leq 12$ that is equivalent to inequality (iii). Hence, (iii) cannot define a facet. □

2.3.3 Inequalities for the polytope of m-sparse triple systems

In this section, we study polytopes for triple systems avoiding subconfigurations.

Let us denote by $P_{AP}(v)$ the polyhedron associated with anti-Pasch 2–$(v,3,1)$ packings and by $P_m(v)$ the polyhedron associated with m-sparse 2–$(v,3,1)$ packings. As we have seen in Section 1.3.4, anti-Pasch and 4-sparse triple systems are equivalent, so $P_{AP}(v) = P_4(v)$.

The main contribution of this section is a class of facet inducing inequalities for $P_{AP}(v)$ given by Theorem 2.3.16 and its generalization for $P_m(v)$ given by Theorem 2.3.20.

Let us start with some basic properties of these polytopes.

Proposition 2.3.15 *(Dimension)*
For $m \geq 4$, the polyhedron $P_m(v) \subseteq R^{\binom{v}{3}}$ is full dimensional.

Proof. Since the vectors $\mathbf{0}$, e_S, for all $S \in \binom{V}{3}$, are incidence vectors of trivially m-sparse 2-$(v,3,1)$ packing designs and are affinely independent. We conclude that $dim P_m(v) = \binom{v}{3}$.
□

Note that although very often the inclusion $P_m(v) \subseteq P_{2,v,3,1}$ is proper, the maximal clique inequalities continue to induce facets for $P_m(v)$. This is due to the fact that in the proof of facetness of maximal clique inequalities, independent sets with at most 2 vertices are employed; in the context of our polytopes, they clearly correspond to m-sparse Steiner triple systems and so the proof remains valid.

For the sake of clarity, we first exhibit a class of facet-defining inequalities for the polytope $P_{AP}(v)$, and then generalize it for $P_m(v)$, for $m \geq 4$.

Let us define a *quasi-Pasch* to be any set of 3 distinct triples contained in some Pasch configuration. Given a valid inequality $d^T x \leq d_0$ for a polytope P, we define its equality set as $EQ(d^T x \leq d_0) = \{x \in P : d^T x = d_0\}$.

Theorem 2.3.16 *(Pasch-avoiding facets)*

Let T be a 6-set contained in $[1, v]$. Then the inequality

$$pa(T) : \quad \sum_{S \in \binom{T}{3}} x_S \leq 3$$

defines a facet for $P_{AP}(v)$. We call the inequality $pa(T)$ the Pasch-avoiding inequality relative to T.

Proof. Throughout the proof we will assume w.l.o.g. that $T = [1, 6]$.

1. The inequality $pa(T)$ is valid.

   If we use the fact that $P_{AP}(v) = P_4(v)$ the argument would be simpler (see Theorem 2.3.20- Part 1.). The following lines actually show that anti-Pasch packings are 4-sparse.

   We just have to show that no more than 3 blocks from $\binom{T}{3}$ can be present in an anti-Pasch 2-$(v, 3, 1)$ packing design, say $(V, \mathcal{B})$. Suppose we have at least 2 distinct blocks in $\mathcal{B}$ from $\binom{T}{3}$. Call these two blocks B_1 and B_2. Then, they either intersect in zero or one point. If they intersect in zero points then no other block of the design can be in $\binom{T}{3}$, for such a block would necessarily have a pair in common with either B_1 or B_2. The last case to examine is $|B_1 \cap B_2| = 1$. We assume w.l.o.g. that $B_1 = \{1, 2, 3\}$ and $B_2 = \{1, 4, 5\}$. From the packing design property, the only candidates from $\binom{T}{3}$ that can be in $\mathcal{B}$ are precisely in $\mathcal{C} = \{\{2, 4, 6\}, \{2, 5, 6\}, \{3, 4, 6\}, \{3, 5, 6\}\}$. Finally, we see that if more than one set from $\mathcal{C}$ would be in $\mathcal{B}$, either the packing or the anti-Pasch properties would be violated. Therefore, $\left|\mathcal{B} \cap \binom{T}{3}\right| \leq 3$.

2. The inequality $pa(T)$ defines a facet.

   It is enough to prove that if $a^T x \leq \alpha$ is a valid inequality for $P_{AP}(v)$ with $EQ(a^T x \leq \alpha) \supseteq EQ(pa(T))$, then $a^T x \leq \alpha$ is a scalar multiple of $pa(T)$ (see Theorem 1.2.4).

Let S and S' be arbitrary sets in $\binom{T}{3}$ such that $|S \cap S'| = 2$. Let us assume w.l.o.g. that $S = \{1,2,3\}$ and $S' = \{1,2,4\}$. There exists a pair of sets $S_1, S_2 \in \binom{T}{3}$ such that both $\mathcal{B}_1 = \{S_1, S_2, S\}$ and $\mathcal{B}_2 = \{S_1, S_2, S'\}$ are quasi-Pasches, e.g., $S_1 = \{3,4,5\}$ and $S_2 = \{1,5,6\}$. Clearly $(V, \mathcal{B}_1)$ and $(V, \mathcal{B}_2)$ are anti-Pasch 2-$(v,3,1)$ packings. Letting x and x' be their corresponding incidence vectors, we observe that $x = x' - e_{S'} + e_S$. Moreover, we have $x, x' \in EQ(pa(T)) \subseteq EQ(a^T x \leq \alpha)$. Thus, $\alpha = a^T x' = a^T x = a^T(x' - e_{S'} + e_S)$, which implies $a_{S'} = a_S$. We claim that there exists a constant γ such that $a_S = \gamma$, for all $S \in \binom{T}{3}$. Indeed, this follows trivially from the connectivity of the graph whose vertices correspond to the sets in $\binom{T}{3}$ and such that two sets are joined by an edge whenever their intersection has size 2.

Next we will show that $a_R = 0$, for all $R \in \binom{V}{3} \setminus \binom{T}{3}$. Let $R \in \binom{V}{3} \setminus \binom{T}{3}$. We claim that there exists a quasi-Pasch in $\binom{T}{3}$ that together with R forms an anti-Pasch 2-$(v,3,1)$ packing. If $|R \cap T| \leq 1$, then any quasi-Pasch with sets in $\binom{T}{3}$ satisfy this property. So, let us analyze the remaining case of $|R \cap T| = 2$. In this case, we assume w.l.o.g. that $R = \{1,2,7\}$. Thus, the following quasi-Pasch $\mathcal{B}_1 = \{\{1,3,4\},\{1,5,6\},\{2,3,5\}\}$ has the desired property. Therefore, both $\mathcal{B}_1 \cup \{R\}$ and $\mathcal{B}_1$ are anti-Pasch 2-$(v,3,1)$ packings. Moreover, their corresponding incidence vectors, say x and x', are in $EQ(pa(T)) \subseteq EQ(a^T x \leq \alpha)$. Therefore, we conclude that $\alpha = a^T x' = a^T x = a^T(x' + e_R) = a^T x' + a_R$, and so $a_R = 0$.

We have shown that $a^T x \leq \alpha$ can be rewritten as $\gamma \sum_{S \in \binom{T}{3}} x_S \leq \alpha$. Letting $\overline{x}$ be any quasi-Pasch with sets in $\binom{T}{3}$, we know that $\overline{x} \in EQ(pa(T)) \subseteq EQ(\gamma \sum_{S \in \binom{T}{3}} x_S \leq \alpha)$, and one can easily derive that $\alpha = 3\gamma$. Therefore, $pa(T)$ and $a^T x \leq \alpha$ only differ by the multiplication of the scalar γ.

□

In order to generalize Theorem 2.3.16 for m-sparse 2–$(v,3,1)$ packings, we need some definitions and two auxiliary lemmas.

Following the definition in [46], an *Erdös configuration of order* n, $n \geq 1$, in a Steiner triple system is any $(n+2, n)$-configuration, which contains no $(l+2, l)$-configuration,

$1 < l < n$. In fact, this is equivalent to requiring that $4 \leq l < n$, since there cannot be any $(4,2)$- or $(5,3)$-configurations in a STS.

Lemma 2.3.17 *(Lefmann et al.[63, Lemma 2.3])*
Let l, r be positive integers, $l \geq 1$. Then any $(l+2, l+r)$-configuration in a Steiner triple system contains a $(l+2, l)$-configuration.

Lemma 2.3.18 *(Construction of an Erdös configuration, for all $n \geq 4$)*
Consider the following recursive definition:

$$\begin{aligned}
\mathcal{E}_4 &= \{E_1 = \{1,2,5\}, E_2 = \{3,4,5\}, E_3 = \{1,3,6\}, E_4 = \{2,5,6\}\}, \\
\mathcal{E}_5 &= \mathcal{E}_4 \setminus \{E_4\} \cup \{\{2,5,7\}, E_5 = \{5,6,7\}\} \\
\mathcal{E}_{n+1} &= \mathcal{E}_n \setminus \{E_n\} \cup \{E_n \setminus \{n+2\} \cup \{n+3\}\} \cup \\
&\quad \cup \{E_{n+1} = \{n+2, n+3, 1 + ((n-1) \bmod 4)\}\}, n > 5.
\end{aligned}$$

Then, for all $n \geq 4$, $\mathcal{E}_n$ is an Erdös configuration of order n.

Proof. $\mathcal{E}_4$ is the Pasch configuration, which is the only $(6,4)$-configuration. $\mathcal{E}_5$ is the mitre configuration, which is the only $(7,5)$-configuration not containing a Pasch. Assume that $\mathcal{E}_n$ is an Erdös configuration of order n. We will prove by induction on n that $\mathcal{E}_{n+1}$ is an Erdös configuration of order $n+1$. $\mathcal{E}_{n+1}$ is clearly an $(n+3, n+1)$-configuration. It remains to show that it does not contain any $(l+2, l)$-configuration for $l \leq n$. Suppose to the contrary that there exists an $(l+2, l)$-configuration $\mathcal{B}$ in $\mathcal{E}_{n+1}$, with $4 \leq l \leq n$. First, we observe that $\mathcal{B}$ must contain one of the 2 blocks containing the last point $n+3$, for otherwise $\mathcal{B}$ would be contained in $\mathcal{E}_n$. In fact, $\mathcal{B}$ must contain both blocks containing $n+3$, for if contains only one, call it E, $\mathcal{B} \setminus \{E\} \subseteq \mathcal{E}_n$ would be an $(p, l-1)$-configuration with $l-1 \leq p \leq l+1$, and by Lemma 2.3.17, it would contain a $(p, p-2)$-configuration, which is contrary to our assumption.

Let x be the maximum number for which the block $E_x \in \mathcal{E}_{n+1}$ does not appear in $\mathcal{B}$. Since $x \leq n-1$ and $l \leq n$, we must have $x \geq 1$. So, $E_{x+1}, \ldots, E_{n+1} \in \mathcal{B}$ and $E_x \notin \mathcal{B}$, for some $1 \leq x \leq n-1$. By construction, $E_{x+1}, E_{x+2}, \ldots, E_{n+1}$ are the only blocks containing $\{x+3, x+4, \ldots, n+3\}$. Let $\mathcal{B}' = B \setminus \{E_{x+1}, E_{x+2}, \ldots, E_{n+1}\}$. Then, $\mathcal{B}' \subseteq \mathcal{E}_n$

has $l-(n-x)-1$ blocks and span $p \leq l+2-(n-x)-1$ points. We observe that $p \geq 3$, since we are dealing with at least one triple. Then, by Lemma 2.3.17, $B' \subseteq \mathcal{E}_n$ contains an $(p, p-2)$ configuration which is a contradiction. $\square$

Lemma 2.3.19 *Let $v \geq l \geq 4$ and let T be an $(l+2)$-subset of $[1,v]$. Let $R \in \binom{[1,v]}{3}$, $R \not\subseteq T$. Then, there exists an Erdös configuration $\mathcal{S}$ of order l on the points of T and a triple $S \in \mathcal{S}$, such that $\mathcal{S} \setminus \{S\} \cup \{R\}$ is an l-sparse 2–$(v,3,1)$ packing.*

Proof. Let $\mathcal{S}$ be an Erdös configuration of order l on the points of T (Lemma 2.3.18 guarantees its existence). If $|R \cap T| \leq 1$, taking any $S \in \mathcal{S}$, the set $\mathcal{S}' = \mathcal{S} \setminus \{S\} \cup \{R\}$ is a 2–$(v,3,1)$ packing. Otherwise, if $|R \cap T| = 2$, we will choose $\mathcal{S}$ such that the pair $R \cap T$ appears in a block, say S. Then, $\mathcal{S}' = \mathcal{S} \setminus \{S\} \cup \{R\}$ is a 2–$(v,3,1)$ packing. In either case, we claim $\mathcal{S}'$ does not contain an $(n+2,n)$-configuration, $4 \leq n \leq l$. Indeed, if that was the case, the configuration, say $\mathcal{B}$, would contain R. Moreover, since $R \not\subseteq T$ and $|R| = 3$, $\mathcal{B} \setminus \{R\}$ would be a $(p, n-1)$-configuration with $n-1 \leq p \leq n+1$. Thus, by Lemma 2.3.17, $\mathcal{B} \subseteq \mathcal{S}$ would contain a $(p,p-2)$-configuration for $p \geq 3$, which is a contradiction. $\square$

Theorem 2.3.20 *(m-sparse facets)*
For any $4 \leq l \leq m$ and any $(l+2)$-subset T of $[1,v]$, the inequality

$$s(T): \quad \sum_{S \in \binom{T}{3}} x_S \leq l-1$$

defines a facet for $P_m(v)$.

Proof.

1. The inequality $s(T)$ is valid.

   Let $(V, \mathcal{B})$ be an m-sparse 2–$(v,3,1)$ packing. Let $\mathcal{B}_T$ be $\mathcal{B} \cap \binom{T}{3}$ and let U be the union of all the triples in B_T. Let $p = |U|$. Clearly $p \leq |T| = l+2$. Since $(V, \mathcal{B})$ is m-sparse, then $|\mathcal{B}_T| \leq p-1$. Thus, $|\mathcal{B}_T| \leq (l+2)-1 = l-1$, and so $s(T)$ is valid.

2. The inequality $s(T)$ defines a facet.

Let $a^T x \leq \alpha$ be a valid inequality for $P_m(v)$ with $EQ(a^T x \leq \alpha) \supseteq EQ(s(T))$. We must show that $a^T x \leq \alpha$ and $s(T)$ are scalar multiples of each other.

Let $\mathcal{S}$ be an Erdös configuration of order l. There must be two triples in $\mathcal{S}$ whose intersection is a single point, call those triples S_1 and S_2. We claim $\mathcal{S} \setminus \{S_1\}$ and $\mathcal{S} \setminus \{S_2\}$ are m-sparse 2-$(v,3,1)$ packings. Indeed, $|\mathcal{S} \setminus \{S_i\}| = |\mathcal{S}| - 1 = l - 1$, and since $\mathcal{S}$ was $(l-1)$-sparse, so is $\mathcal{S} \setminus \{S_i\}$, $i = 1, 2$. Letting x^i be the incidence vector of $\mathcal{S} \setminus \{S_i\}$, for $i = 1, 2$. Then, $x^1 = x^2 - e_{S_2} + e_{S_1}$. In addition, x^1,$x^2 \in EQ(s(T)) \subseteq EQ(a^T x \leq \alpha)$. Thus, $\alpha = a^T x^2 = a^T x^1 = a^T(x^2 - e_{S_2} + e_{S_1})$, which implies $a_{S_1} = a_{S_2}$. Moreover, for any $R_1, R_2 \in \binom{T}{3}$ with $|R_1 \cap R_2| = 1$, we can show that $a_{R_1} = a_{R_2}$ by the same argument as before, just by considering an appropriate permutation on T that takes R_i to S_i, $i = 1, 2$. Consider the graph whose vertices are the triples in T and two triples are joined by an edge whenever their intersection has size 1. Since this graph is connected, we conclude that there exists a constant γ such that $a_S = \gamma$, for all $S \in \binom{T}{3}$.

Next we show that $a_R = 0$, for all $R \in \binom{V}{3} \setminus \binom{T}{3}$.

Let $R \in \binom{V}{3} \setminus \binom{T}{3}$. By Lemma 2.3.19, there exists an Erdös configuration of order l, say $\mathcal{S}$ and a triple $S \in \mathcal{S}$, such that $\mathcal{S} \setminus \{S\} \cup \{R\}$ is an m sparse 2-$(v,3,1)$ packing. Let x and x_R be the incidence vectors of $\mathcal{S} \setminus \{S\}$ and $\mathcal{S} \setminus \{S\} \cup \{R\}$, respectively. Then $x, x_R \in EQ(s(T)) \subseteq EQ(a^T x \leq \alpha)$, and thus $\alpha = a^T x = a^T x_R = a^T(x + e_R) = a^T x + a_R$, and so $a_R = 0$.

So, similarly to the proof of Theorem 2.3.16, we conclude $s(T)$ is facet defining.

□

Corollary 2.3.21 *(Separation of m-sparse facets)*

For constant $m \geq 4$, m-sparse facets can be separated in polynomial time.

Proof. The number of such facets is $\sum_{l=4}^{m} \binom{v}{l}$ which, for constant m, is bounded by a polynomial on v. □

The proof of Corollary 2.3.21 implies the use of the algorithm that checks whether each of the possible inequalities is violated. In Chapter 4, more efficient algorithms are given for the separation of m-sparse facets for some small m's.

2.3.4 Adjacency in the polytope $T_{t,v,k,1}$ and null designs

A *null* (t, v, k)-design is a $(0, 1, -1)$-vector satisfying $W_{t,k}^{v} x = 0$. Let x^1, x^2 be incidence vectors of t-(v, k, λ) designs. Then, $W_{t,k}^{v}(x^1 - x^2) = \lambda - \lambda = 0$, so $x^1 - x^2$ corresponds to a null design.

Let let $k \geq t+1$, $\overline{K} \in \binom{[1,v]}{k}$ and $T \in \binom{[1,v]}{t+1}$ with $\overline{K} \cap T = \emptyset$, and ϕ be an injection from T to $\overline{K}$. Define $y \in \mathbb{R}^{\binom{v}{k}}$ such that $y_K = (-1)^{|K \cap T|}$ whenever $K \subseteq \overline{K} \cup T$ and K contains exactly one of each pair $(i, \phi(i))$ for all $i \in T$, and $y_K = 0$, otherwise. Such y is called a (t, k)-*pod* and it is known to be a null (t, v, k)-design with minimum support (see [44]).

For example, consider a $(2, 3)$-pod given by $\overline{K} = \{1, 2, 3\}$ $T = \{4, 5, 6\}$ with $\phi(6) = 1$, $\phi(5) = 2$ and $\phi(4) = 3$. Consider the following triples:

$\mathcal{A}$	$\mathcal{B}$
{1,2,3}	{6,2,3}
{1,4,5}	{6,4,5}
{2,4,6}	{2,4,1}
{3,5,6}	{3,5,1}

The $(2, 3)$-pod is given by $y_K = 1$ for $K \in \mathcal{A}$, $y_K = -1$ for $K \in \mathcal{B}$ and $y_K = 0$ otherwise. The four triples in $\mathcal{A}$ and $\mathcal{B}$ both cover all pairs from $[1, 6]$ except the pairs $\{1, 6\}$, $\{2, 5\}$ and $\{3, 4\}$; they form a pair of Pasches. Thus, if a STS, say x^1, contains the Pasch $\mathcal{B}$, then $x^2 = x^1 + y$ is a STS with $\mathcal{B}$ replaced by $\mathcal{A}$. The next theorem shows that if two t–$(v, k, 1)$ designs x^1 and x^2 differ by a (t, k)-pod, then they are adjacent vertices of the polytope $T_{t,v,k,1}$.

Theorem 2.3.22 *Let x^1 and x^2 be incidence vectors of t–$(v, k, 1)$ designs for which $x^1 - x^2$ is a (t, k)-pod. Then x^1 and x^2 are adjacent vertices in the polytope $T_{t,v,k,1}$.*

Proof. By definition x^1 and x^2 are vertices of $T_{t,v,k,1}$ and $P_{t,v,k,1}$. Using Theorem 2.2.8, x^1 and x^2 are adjacent if and only $G^v_{t,k}[\text{supp}(x^1 - x^2)]$ is connected. Let $K_1, K_2 \in \text{supp}[x^1 - x^2]$. We prove by induction on $l = |K_1 \setminus K_2| = |K_2 \setminus K_1|$ that there exist a path connecting K_1 and K_2. If $l = 1$, K_1 and K_2 differ by one element, so $|K_1 \cap K_2| = k-1 \geq t$, and there is an edge connecting K_1 and K_2 in $G^v_{t,k}$. If $l > 1$, then take $s \in K_1 \setminus K_2$. By definition, $\phi(s) \in K_2$. Take $K_2' = K_2 \setminus \{\phi(s)\} \cup \{s\}$, and observe that by the definition of a (t,k)-pod $K_2' \in \text{supp}(x^1 - x^2)$. Then $|K_1 \setminus K_2'| = l - 1$ and $|K_2 \setminus K_2'| = 1$ and by induction, there exists a path in $G^v_{t,k}[\text{supp}(x^1 - x^2)]$ connecting K_1 and K_2' as well as K_2 and K_2'. Therefore, there exists a path in $G^v_{t,k}[\text{supp}(x^1 - x^2)]$ connecting K_1 and K_2. □

The condition in the previous theorem is not necessary, as will be seen in the example of Section 2.4.1.

2.4 Complete descriptions of some polytopes

We conclude this chapter by studying in more detail the Fano plane polytope (i.e. the polytope $T_{2,7,3,1}$) and showing all the facets for the packing design polytopes with small parameters ($t = 2$, $k = 3$ and $v \leq 5$).

2.4.1 The Fano plane polytope or $T_{2,7,3,1}$

The vertices of the polytope $T_{2,7,3,1}$ are the incidence vectors of the $2-(7,3,1)$ designs. These designs are symmetric since the number of blocks is equal to the number of points. They are also known to correspond to projective planes of order 2. Table 2.2 shows these designs. One can classify each pair of such designs as one of 3 types: the pairs that intersect in 3 blocks, in 1 block, and in no block. Figure 2.3 displays these intersections. G1 is the union of two $\mathbb{K}_{15}$ with edges connecting designs that intersect in 1 block; G2 is a bipartite graph with edges connecting designs with 2 blocks in common; G3 is the complement of G1 ∪ G2 (the missing edges in the bipartite graph) and its edges connect designs with no common blocks.

It turns out that every pair of the design's incidence vectors are linked by an edge, i.e., a face of dimension 1, in the polytope $T_{2,7,3,1}$. This can be verified by examining every

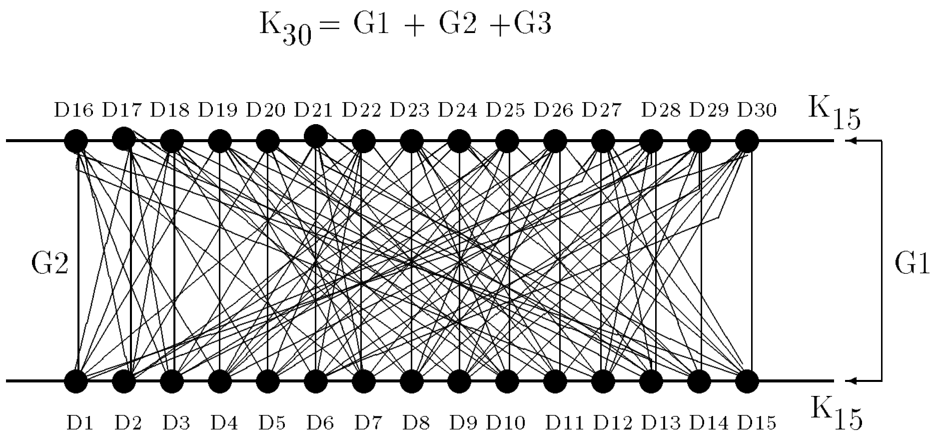

Figure 2.3: The three types of intersection among $2-(7,3,1)$ designs

pair of distinct designs (D_i, D_j). If $\mathcal{B}_i, \mathcal{B}_j$ are their respective set of blocks, we observe that $(\mathcal{B}_i \setminus \mathcal{B}_j) \cup (\mathcal{B}_j \setminus \mathcal{B}_i)$ induces a connected subgraph of $W^7_{2,3}$ and by Theorem 2.2.8, there is an edge between D_i and D_j in $P_{2,7,3,1}$. Since $T_{2,7,3,1}$ is a face of $P_{2,7,3,1}$, there is an edge connecting D_i and D_j in $T_{2,7,3,1}$.

Other properties of this polytope were determined using the Porta software by Thomas Christof and Andreas Loebel [20]. For the 30 vertices of $T_{2,7,3,1}$ the software computed its equality and inequality representation. The information obtained is summarized in the following:

- $T_{2,7,3,1} \in \mathbb{R}^{35}$ has dimension 14 (its dimension achieves the upper bound of Theorem 2.3.1). Thus, its equality system is given solely by $W^7_{2,3}x = \mathbf{1}$.

- Its inequality system contains 155 facet-inducing inequalities of two types. Thirty five of these inequalities are equivalent to $x_i \geq 0$, for $i = 1, \ldots, 35$. Each of these inequalities passes through exactly 24 vertices. Each of the others 120 inequalities passes through exactly 14 vertices.

- There are 84 inequalities passing through each vertex.

D_1	123	145	167	246	257	347	356	D_{16}	123	145	167	247	256	346	357
D_2	123	146	157	247	256	345	367	D_{17}	123	146	157	245	267	347	356
D_3	123	147	156	245	267	346	357	D_{18}	123	147	156	246	257	345	367
D_4	124	135	167	237	256	346	457	D_{19}	125	134	167	237	246	356	457
D_5	124	136	157	235	267	347	456	D_{20}	126	134	157	235	247	367	456
D_6	124	137	156	236	257	345	467	D_{21}	127	134	156	236	245	357	467
D_7	125	134	167	236	247	357	456	D_{22}	124	135	167	236	257	347	456
D_8	125	136	147	237	246	345	567	D_{23}	126	135	147	237	245	346	567
D_9	125	137	146	234	267	356	457	D_{24}	127	135	146	234	256	367	457
D_{10}	126	134	157	237	245	356	467	D_{25}	124	136	157	237	256	345	467
D_{11}	126	135	147	234	257	367	456	D_{26}	125	136	147	234	267	357	456
D_{12}	126	137	145	235	247	346	567	D_{27}	127	136	145	235	246	347	567
D_{13}	127	134	156	235	246	367	457	D_{28}	124	137	156	235	267	346	457
D_{14}	127	135	146	236	245	347	567	D_{29}	125	137	146	236	247	345	567
D_{15}	127	136	145	234	256	357	467	D_{30}	126	137	145	234	257	356	467

Table 2.2: The 30 distinct 2-$(7,3,1)$ designs.

2.4.2 Polyhedra for small packings

Next we list the complete facet representation of polytopes for some small packings, obtained using Porta software [20].

$$P_{2,4,3,1}: \begin{cases} \sum_{K\in\binom{[1,4]}{3}} x_K \leq 1, & \\ x_K \geq 0, & \text{for all } K \in \binom{[1,4]}{3}. \end{cases}$$

$$P_{2,5,3,1}: \begin{cases} \sum_{K\supseteq T} x_K \leq 1, & \text{for all } T \in \binom{[1,5]}{2} \\ \sum_{K\subseteq F} x_K \leq 1, & \text{for all } F \in \binom{[1,5]}{4} \\ \sum_{K\in\binom{[1,5]}{3}} x_K \leq 2, & \\ x_K \geq 0, & \text{for all } K \in \binom{[1,5]}{3}. \end{cases}$$

$$P_{2,5,3,2}: \begin{cases} x_K \leq 1, & \text{for all } K \in \binom{[1,5]}{3}, \\ \sum_{K\supseteq T} x_K \leq 2, & \text{for all } T \in \binom{[1,5]}{2} \\ \sum_{K\in\binom{[1,5]}{3}} x_K \leq 6, & \\ x_K \geq 0, & \text{for all } K \in \binom{[1,5]}{3}. \end{cases}$$

Chapter 3

Clique Facets and Intersecting Set Systems

As shown in Chapter 2, the generalized clique facets correspond to set systems with prescribed intersection properties (see Proposition 2.3.6 and Corollary 2.3.7). In this chapter, we take a closer look at these intersecting set systems. Our main goal is the classification of all the clique facets of $P_{t,v,k,\lambda}$, which are precisely the facets with right-hand side being equal to 1.

We recall some definitions from Chapter 2. Given $s \geq 2$ and $v, t \geq 1$, a family $\mathcal{A}$ of subsets of $[1, v]$ is said to be *s-wise t-intersecting*, if any s members $A_1, \ldots, A_s$ of $\mathcal{A}$ are such that $|A_1 \cap \ldots \cap A_s| \geq t$. A family $\mathcal{A}$ is said to be *k-uniform* if every member of $\mathcal{A}$ has cardinality k. We denote by $I^s(v, k, t)$ the set of all k-uniform s-wise t-intersecting families of subsets of $[1, v]$. We are interested in the families in $I^s(v, k, t)$ that are maximal with respect to set inclusion, i.e., the families $\mathcal{A} \in I^s(v, k, t)$ such that for any $\mathcal{B} \in I^s(v, k, t)$, if $\mathcal{B} \supseteq \mathcal{A}$ then $\mathcal{B} = \mathcal{A}$. We denote by $MI^s(v, k, t)$ the set of all families in $I^s(v, k, t)$ that are maximal in this sense. The problem that motivates this chapter is the following.

Problem 1. Given $\lambda \geq 1$ and $v > k > t$, list all generalized maximal clique facets of $P_{t,v,k,\lambda}$.

By Proposition 2.3.6, this problem can be rewritten in terms of intersecting families of sets, taking $s = \lambda + 1$, as follows.

Problem 2. Given $s \geq 2$, $v > k > t$, list all families in $MI^s(v, k, t)$; or alternatively, list all nonisomorphic families in $MI^s(v, k, t)$.

In addition to the above mentioned polyhedral consequences, Problem 2 is of independent interest in extremal set theory and in the theory of hypergraphs. We briefly comment on related works. The famous Erdös-Ko-Rado theorem [33] determined the size of the largest family in $MI^2(v, k, t)$ for v large enough, and it was a seminal paper in the study of intersecting families. The best possible v for which this theorem remains true was shown by Frankl [36], for $t \geq 15$, and by Wilson [105], for general t. Other extensions of the Erdös-Ko-Rado theorem includes works of Hilton and Milner [49] and Frankl [38]. Recently, the problem of determining the largest family in $MI^2(v, k, t)$ was finally settled for the whole spectrum of v by Ahlswede and Khachatrian [2] [3].

Problem 2 involves determining all the families in $MI^s(v, k, t)$, not just the largest ones, which in general turns out to be a hard problem. Nevertheless, some partial results on the classification of the families in $MI^s(v, k, t)$ are presented in this chapter; the most important ones are:

1. the classification problem is solved for the first nontrivial case, namely $s = 2$ and $k = t + 2$;

2. for any s, k, t, the classification problem does not depend on v.

Other intermediate and interesting results are shown along the way and described in more detail in the next section.

The results in this chapter are contained in paper [77]. In another work, we use collections of intersecting set systems for studying transversal covers [94].

3.1 Definitions and results

In this chapter, we make use of two important concepts in the study of families in $MI^s(v, k, t)$, namely kernels and generating sets. First, we define the notion of a generating set of a family of subsets, as introduced by Ahlswede and Khachatrian [2] [3]. Let us fix some notation. For any $B \subseteq [1,v]$, let $\mathcal{U}_v^k(B) = \{C \subseteq [1,v] : C \supseteq B, |C| = k\}$, and for a family $\mathcal{B}$ of subsets of $[1,v]$, let $\mathcal{U}_v^k(\mathcal{B}) = \cup_{B \in \mathcal{B}} \mathcal{U}_v^k(B)$. Let $\mathcal{A}$ be a k-uniform family of subsets of $[1,v]$; a family g of subsets of $[1,v]$ is called a *generating set* of $\mathcal{A}$, if $\mathcal{U}_v^k(g) = \mathcal{A}$. The set $\mathcal{A}$ itself is a generating set of $\mathcal{A}$, although not a very interesting one. We would like to be able to find generating sets g that are "minimal" in some sense, for example, the ones for which $|\cup_{G \in g} \ G|$ or $|g|$ are minimal. Such generating sets provide us with a compact way to represent families of subsets, which proves to be useful for studying Problem 2.

We turn our attention to the concept of a kernel of an s-wise t-intersecting family of subsets. For a family $\mathcal{A}$ in $I^s(v,k,t)$, a set $\mathcal{K} \subseteq [1,v]$ is a *kernel* of $\mathcal{A}$ if any s sets in $\mathcal{A}$ meet in at least t elements of $\mathcal{K}$, i.e., for any $A_1, \ldots, A_s \in \mathcal{A}$ we have $|A_1 \cap \ldots \cap A_s \cap \mathcal{K}| \geq t$. Denote by $n(k,s,t)$ the smallest integer for which every family $\mathcal{A}$ in $\bigcup_v I^s(v,k,t)$ has a kernel $\mathcal{K}(\mathcal{A})$ with $|\mathcal{K}(\mathcal{A})| \leq n(k,s,t)$ (the standard definition of $n(k,s,t)$ does not require the families $\mathcal{A}$ to be k-uniform, but rather having members with cardinality at most k; both definitions turn out to be equivalent). It is not obvious that $n(k,s,t)$ is finite for all k, s and t. Całczyńska-Karłowicz [15] proved that $n(k,2,1)$ is finite. The first explicit upper bound on $n(k,2,1)$ was given by Ehrenfeucht and Mycielski [32]. Lower bounds and further improvements on upper bounds were given by Erdös and Lovász [34] and Tuza [101]. The finiteness of $n(k,s,t)$ was independently proven by Frankl [37], in an implicit form, Kahn and Seymour [51], and Frankl and Füredi [39]. These results imply that at least t elements of each of the s-wise intersections of a family $\mathcal{A} \in I^s(v,k,t)$ are included in a small subset (with cardinality independent of v) of the underlying set $[1,v]$. General bounds on $n(k,s,t)$ are given by Alon and Füredi [4], who proved that

$$\frac{1}{3}\left(3^{1/3}\right)^{(k-t)/(s-1)} \leq n(k,s,t) \leq k\binom{k-t+1+\lfloor\frac{k-t}{s-1}\rfloor}{\lfloor\frac{k-t}{s-1}\rfloor} \leq k\left(\frac{s^s}{(s-1)^{s-1}}\right)^{\lfloor(k-t)/(s-1)\rfloor} \tag{3.1}$$

They give an explicit construction for the lower bound, and also show that the upper bound follows from a theorem of Füredi [41]. Kohayakawa [53] sharpened the lower bound in (3.1).

We note that the upper bound in (3.1) can be improved by a simple observation on the last step of Alon and Füredi's proof. Indeed, in that proof [4, Theorem 2.2], they obtain an upper bound on the maximum size of a kernel given by the size of the union of m sets of size k, with $m = \binom{k-t+1+\lfloor\frac{k-t}{s-1}\rfloor}{\lfloor\frac{k-t}{s-1}\rfloor}$. This leads to the upper bound in (3.1). However, the m sets form an s-wise t-intersecting family, which implies that every set has at least t elements in common with a fixed set, say the first one. Therefore, the upper bound can be improved to:

$$n(k,s,t) \leq (k-t)\binom{k-t+1+\lfloor\frac{k-t}{s-1}\rfloor}{\lfloor\frac{k-t}{s-1}\rfloor} + t \leq (k-t)\left(\frac{s^s}{(s-1)^{s-1}}\right)^{\lfloor(k-t)/(s-1)\rfloor} + t \qquad (3.2)$$

Before summarizing the contents of the next sections on this chapter, we give some definitions and fix some notation.

Definition 3.1.1 *We say that two families $\mathcal{A}^1$ and $\mathcal{A}^2$ of subsets of $[1,v]$ are* isomorphic, *denoted by $\mathcal{A}^1 \sim \mathcal{A}^2$, if there exist a permutation π of the set $[1,v]$, such that π sends $\mathcal{A}^1$ to $\mathcal{A}^2$. We also write $\mathcal{A}^2 = \pi(\mathcal{A}^1)$.*

Notation 3.1.2 *Let $\mathcal{C}$ be a collection of families of subsets. We denote by $D(\mathcal{C})$ the number of nonisomorphic members of $\mathcal{C}$, that is, the number of equivalent classes induced by the equivalent relation given by $\sim$.*

Definition 3.1.3 *Let $\mathcal{C}^1, \mathcal{C}^2$ be collections of families of subsets. We say that collections $\mathcal{C}^1$ and $\mathcal{C}^2$ are* isomorphism-equivalent, if for every family $\mathcal{A}^1 \in \mathcal{C}^1$ there exists a family $\mathcal{A}^2 \in \mathcal{C}^2$ such that $\mathcal{A}^2 \sim \mathcal{A}^1$, and vice-versa.

In Section 3.2, we study the relationship between kernels and generating sets of families in $MI^s(v,k,t)$. We introduce the notion of a set of essential elements $E(\mathcal{A}) \subseteq [1,v]$ for a family $\mathcal{A} \in I^s(v,k,t)$. We show that for maximal families the concepts of kernels, generating sets, and essential elements are closely related (Theorem 3.2.4). In particular, for $v > 2k-t$ and $\mathcal{A} \in MI^s(v,k,t)$, we exhibit generating sets $g(\mathcal{A})$ and $g'(\mathcal{A})$ of $\mathcal{A}$ such that $E(\mathcal{A}) = \cup_{G \in g(\mathcal{A})} G = \cup_{G \in g'(\mathcal{A})} G$ is the unique kernel of $\mathcal{A}$ with minimum cardinality. We also show that for v large enough, more precisely, for $v \geq n_0(k,s,t) = \max\{2k-t+1, n(k,s,t)\}$, the set formed by the generating sets $g'(\mathcal{A})$ for all $\mathcal{A} \in MI^s(v,k,t)$ is isomorphism-equivalent to the set formed by the generating sets $g'(\mathcal{B})$ for all $\mathcal{B} \in MI^s(n_0(k,s,t),k,t)$. Therefore, given k, s and t, in order to solve

Problem 2 for $v \geq n_0(k,s,t)$, it is enough to list the generating sets g' for the solutions obtained for $v = n_0(k,s,t)$.

In Section 3.3, we give a general construction for families in $MI^s(v+1,k+1,t+1)$ based on families in $MI^s(v,k,t)$. This construction produces nonisomorphic families when applied to nonisomorphic ones. This implies that $D(MI^s(v+l,k+l,t+l)) \geq D(MI^s(v,k,t))$, for all $l \geq 0$.

In Section 3.4, we solve Problem 2 for $s = 2$ and $k \leq t+2$. The case of $k = t+1$ is simple, but is included for the sake of completeness. The solution of Problem 2 for $k = t+2$ is more interesting. We show that the construction given in Section 3.3, when applied to all sets in $MI^2(v,t+2,t)$, gives all the sets in $MI^2(v+1,t+3,t+1)$, for all v and $t \geq 2$. This reduces the case $k = t+2$ to the determination of families in $MI^2(n_0(4,2,2),4,2)$ and $MI^2(n_0(3,2,1),3,1)$, which were obtained by computer. The upper bound in (3.2) gives $n(t+1,2,t) \leq t+3$ and $n(t+2,2,t) \leq t+20$, but we prove that $n(t+1,2,t) = t+2$ and $n(t+2,2,t) = t+6$, for any $t \geq 1$. More precisely, the results regarding enumeration of nonisomorphic sets for the case of $k-t \leq 2$ are as follows. For any $t \geq 1$ we have

$$D(MI^2(v,t+1,t)) = \begin{cases} 2, & \text{for } v \geq t+3, \\ 1, & \text{for } v \leq t+2. \end{cases}$$

The form of such sets is given in Theorem 3.4.1. For any $v \geq t+6$, we have

$$D(MI^2(v,t+2,t)) = \begin{cases} 15, & \text{for } t = 1, \\ 17, & \text{for } t \geq 2. \end{cases}$$

The generating sets for the corresponding families are given in Table 3.1.

3.2 Kernels and generating sets

We begin this section by introducing the new concept of a number in $[1,v]$ being *essential* for a family in $I^s(v,k,t)$. Then, in Theorem 3.2.4, we show some properties relating kernels, generating sets, and essential numbers for families in $MI^s(v,k,t)$. In Corollary 3.2.5, we rewrite $n(k,s,t)$ based on essential numbers for families in $MI^s(v,k,t)$ with $v > 2k-t$. Then, we exhibit some special generating sets (given in (3.5)) for families in $MI^s(v,k,t)$

with $v \geq n_0(k,s,t) = \max\{2k-t+1, n(k,s,t)\}$, that allows us to show that a solution for Problem 2 for k, s, t and $v \geq n_0(k,s,t)$ can be obtained from the solution for Problem 2 for k, s, t and $v = n_0(k,s,t)$ (Corollary 3.2.10).

Definition 3.2.1 *Let $\mathcal{A} \in I^s(v,k,t)$. We say that $e \in [1,v]$ is* essential *for $\mathcal{A}$, if there exist sets $A_1, \ldots, A_s$ with $e \in A_1 \cap \ldots \cap A_s$ such that $|A_1 \cap \ldots \cap A_s| = t$. The s-tuple $(A_1, \ldots, A_s)$ is called an* essential s-tuple for e in $\mathcal{A}$. *We also say that e is* essential for A_j relatively to $\mathcal{A}$, *for any $1 \leq j \leq s$. The* essential set of $\mathcal{A}$ *is defined by $E(\mathcal{A}) = \{e \in [1,v] : e$ is essential for $\mathcal{A}\}$.*

It is clear from the above definition that every kernel of a family $\mathcal{A} \in I^s(v,k,t)$ must contain $E(\mathcal{A})$. In Theorem 3.2.4, we show that for a family $\mathcal{A} \in MI^s(v,k,t)$ with $v > 2k-t$, the set $E(\mathcal{A})$ is the *unique* kernel of $\mathcal{A}$ with minimum cardinality. We also exhibit a generating set $g(\mathcal{A})$ of $\mathcal{A}$ that is formed by elements in $E(\mathcal{A})$, and that is largely used in the rest of the chapter. For proving Theorem 3.2.4, we need the following two lemmas. The first one generalizes a known result for $s = 2$ (see [3]).

Lemma 3.2.2 *Let $v > 2k-t$. If $\mathcal{A}$ is a family in $I^s(v,k,t)$ and g is a generating set of $\mathcal{A}$, then g is an s-wise t-intersecting family.*

Proof. Suppose that g is not s-wise t-intersecting. Then there exist sets $G_1, \ldots, G_s \in g$ such that $|G_1 \cap \ldots \cap G_s| < t$. We will show k-extensions $S_1, \ldots, S_s$ of $G_1, \ldots, G_s$ such that $|S_1 \cap \ldots \cap S_s| < t$, which is the desired contradiction. Let us call $I = G_1 \cap \ldots \cap G_{s-1}$. We observe that $0 \leq |I| \leq k$, and that $|I \cap G_s| \leq t-1$. Let S_s be a k-extension of G_s such that $|S_s \cap I| \leq t-1$. There exists such an extension, since the numbers available to be in the set S_s amounts to $v - |I| + t - 1 \geq 2k - t + 1 - k + t - 1 = k$. Let $F \subseteq S_s$, $|F| = k-t+1$, $F \cap I = \emptyset$. Make k-extensions $S_1, \ldots, S_{s-1}$ of $G_1, \ldots, G_{s-1}$ such that if $S_i = G_i \cup L_i$, then $L_i \cap F = \emptyset$. This is always possible, since $|F| = k-t+1$, and so $v - |F| \geq 2k-t+1-(k-t+1) = k$. We first observe, since $L_i \cap F = \emptyset$ for $1 \leq i \leq s-1$, that $S_1 \cap \ldots \cap S_{s-1} \cap F = G_1 \cap \ldots \cap G_{s-1} \cap F = I \cap F = \emptyset$. Therefore,

$$|S_1 \cap \ldots \cap S_{s-1} \cap S_s| = |S_1 \cap \ldots \cap S_{s-1} \cap F| + |S_1 \cap \ldots \cap S_{s-1} \cap (S_s \setminus F)| \leq 0 + |S_s \setminus F| = t-1.$$

This is the desired contradiction. $\square$

Lemma 3.2.3 *Let $\mathcal{A}$ be a family in $MI^s(v,k,t)$, and for any $A \in \mathcal{A}$, let $S_A = (A \cap E(\mathcal{A}))$. If $A \in \mathcal{A}$ and $B \subseteq [1,v]$ is such that $B \supseteq S_A$ with $|B| = k$, then $B \in \mathcal{A}$.*

Proof. We will show that $B \in \mathcal{A}$ by induction on $|B \setminus A|$. If $|B \setminus A| = 0$, trivially $B = A \in \mathcal{A}$. If $|B \setminus A| \geq 1$, then choose $x \in B \setminus A$ and $y \in A \setminus B$. Let $B' = (B \setminus \{x\}) \cup \{y\}$. Since $B' \supseteq S_A$ and $|B' \setminus A| < |B \setminus A|$, by induction hypothesis $B' \in \mathcal{A}$. Since y is not essential, $B' \setminus \{y\}$ is t-intersecting with every $(s-1)$-tuple of sets in $\mathcal{A}$, and so is B. Therefore, since $\mathcal{A}$ is maximal, it follows that $B \in \mathcal{A}$. □

The following theorem establishes connections between kernels, generating sets and essential numbers for a family $\mathcal{A}$ in $MI^s(v,k,t)$. In particular, we introduce a generating set formed by essential elements of $\mathcal{A}$, and we show that when $v > 2k - t$ the essential set of $\mathcal{A}$ is the unique kernel of $\mathcal{A}$ with minimum cardinality. We observe that the assumption $v > 2k - t$ is not so restrictive. For example, when $s = 2$, if $v \leq 2k - t$, then the family formed by every k-subset of $[1,v]$ is pairwise t-intersecting, and therefore is the only family in $MI^2(v,k,t)$.

Theorem 3.2.4 *Let $\mathcal{A}$ be a family in $MI^s(v,k,t)$.*

1. *Let us denote by $g(\mathcal{A})$, the family given by*

$$g(\mathcal{A}) = \{A \cap E(\mathcal{A}) : A \in \mathcal{A}\}. \tag{3.3}$$

   Then, $g(\mathcal{A})$ is a generating set for $\mathcal{A}$.

2. *For any kernel $\mathcal{K}$ of $\mathcal{A}$ we have $E(\mathcal{A}) \subseteq \mathcal{K}$.*

3. *If $v > 2k - t$, then:*

   (a) *for any generating set $\overline{g}$ of $\mathcal{A}$, the set $\cup_{G \in \overline{g}} G$ is a kernel of $\mathcal{A}$;*

   (b) *the set $E(\mathcal{A})$ is a kernel of $\mathcal{A}$; moreover, $E(\mathcal{A})$ is the unique kernel of $\mathcal{A}$ with minimum cardinality.*

Proof. Part 1. Trivially $\mathcal{A} \subseteq \mathcal{U}_v^k(g(\mathcal{A}))$, so it remains to prove that $\mathcal{U}_v^k(g(\mathcal{A})) \subseteq \mathcal{A}$. Take $B \in \mathcal{U}_v^k(g(\mathcal{A}))$. Then $B = S \cup L$ with $S \in g(\mathcal{A})$ and $L \subseteq [1,v] \setminus S$. Since $S \in g(\mathcal{A})$, there exists $A \in \mathcal{A}$ with $S = A \cap E(\mathcal{A})$. Therefore, since $B \supseteq S = A \cap E(\mathcal{A})$, by Lemma 3.2.3, $B \in \mathcal{A}$.

Part 2. Let $e \in E(\mathcal{A})$ and $\mathcal{K}$ any kernel of $\mathcal{A}$. Since e is essential for $\mathcal{A}$ there exist sets $A_1, \ldots, A_s \in \mathcal{A}$ such that $e \in A_1 \cap \ldots \cap A_s$ and $|A_1 \cap \ldots \cap A_s| = t$. In addition, since $\mathcal{K}$ is a kernel of $\mathcal{A}$, $|\mathcal{K} \cap A_1 \cap \ldots \cap A_s| \geq t$ and so $e \in \mathcal{K}$.

Part 3a. Let $\overline{g}$ be a generating set of $\mathcal{A}$. Since $v \geq 2k - t$, it follows by Lemma 3.2.2 that $\overline{g}$ is an s-wise t-intersecting family. Let $\mathcal{K} = \cup_{G \in \overline{g}} G$. Let $A_1, \ldots, A_s$ be arbitrary sets in $\mathcal{A}$. Since $\overline{g}$ is a generating set, there exist sets $G_1, \ldots, G_s \in \overline{g}$, with $G_i \subseteq A_i$ for $1 \leq i \leq s$. So, we have

$$\begin{aligned} |\mathcal{K} \cap A_1 \cap \ldots \cap A_s| &\geq |\mathcal{K} \cap G_1 \cap \ldots \cap G_s| \\ &= |G_1 \cap \ldots \cap G_s| \qquad \text{(by construction of } \mathcal{K}) \\ &\geq t \qquad \text{(since } \overline{g} \text{ is } s\text{-wise } t\text{-intersecting).} \end{aligned}$$

Therefore, the set $\mathcal{K}$ is a kernel of $\mathcal{A}$.

Part 3b. In Part 1, we showed that $g(\mathcal{A})$ is a generating set of $\mathcal{A}$. By construction, we see that $E(\mathcal{A}) = \cup_{G \in g(\mathcal{A})} G$. Since $v \geq 2k - t$, it follows by Part 3a that $E(\mathcal{A})$ is a kernel of $\mathcal{A}$. By Part 2, for any $\mathcal{K}$ kernel of $\mathcal{A}$, $E(\mathcal{A}) \subseteq \mathcal{K}$, so $E(\mathcal{A})$ is the unique kernel of $\mathcal{A}$ with minimum cardinality. □

A consequence of this theorem is that we can rewrite $n(k, s, t)$ in terms of essential sets of families in $MI^s(v, k, t)$ with $v > 2k - t$.

Corollary 3.2.5 *Let $k \geq t \geq 1$, $s \geq 2$, and let $E_{max}(k, s, t)$ be given by*

$$E_{max}(k, s, t) = \max_{v > 2k-t, \mathcal{A} \in MI^s(v,k,t)} |E(\mathcal{A})|. \tag{3.4}$$

Then, $n(k, s, t) = E_{max}(k, s, t)$.

Proof. The observations below will be used in the proof:

1. $I^s(v', k, t) \subseteq I^s(v, k, t)$, for any $v \geq v'$.

2. Let $\mathcal{A}, \mathcal{A}' \in I^s(v, k, t)$, $\mathcal{A}' \subseteq \mathcal{A}$. If $\mathcal{K}$ is a kernel of $\mathcal{A}$, then $\mathcal{K}$ is a kernel of $\mathcal{A}'$.

3. Let $\mathcal{A}, \mathcal{A}' \in I^s(v, k, t)$, $\mathcal{A}' \subseteq \mathcal{A}$. If $\mathcal{K}'$ is a kernel of $\mathcal{A}'$, then there exist $\mathcal{K} \supseteq \mathcal{K}'$ such that $\mathcal{K}$ is a kernel of $\mathcal{A}$.

Using the above observations, along with part 3b on Theorem 3.2.4, we have:

$$\begin{aligned} n(k,s,t) &= \max_{v\geq k, \mathcal{A}\in I^s(v,k,t)} \min_{\mathcal{K} \text{ kernel of } \mathcal{A}} |\mathcal{K}|, \quad (\text{by definition of } n(k,s,t)) \\ &= \max_{v>2k-t, \mathcal{A}\in I^s(v,k,t)} \min_{\mathcal{K} \text{ kernel of } \mathcal{A}} |\mathcal{K}|, \quad (\text{by observation 1}) \\ &= \max_{v>2k-t, \mathcal{A}\in MI^s(v,k,t)} \min_{\mathcal{K} \text{ kernel of } \mathcal{A}} |\mathcal{K}|, \quad (\text{by observations 2 and 3}) \\ &= \max_{v>2k-t, \mathcal{A}\in MI^s(v,k,t)} |E(\mathcal{A})|, \quad (\text{by part 3b in Theorem 3.2.4}) \\ &= E_{max}(k,s,t). \end{aligned}$$

□

The rest of this section deals with finding a special generating set for families in $MI^s(v,k,t)$ that is also a generating set for families in $MI^s(v',k,t)$, for any $v' \geq v$. The following is an example of a family $\mathcal{A} \in MI^2(8,4,1)$ for which $g(\mathcal{A})$ is not such a special generating set. For $v' \geq 9$, the family $\mathcal{U}^4_{v'}(g(\mathcal{A}))$ is not maximal.

Example 3.2.6 *Let* $g^1 = \{\{1,2,3,4\},\{1,5,6,7\},\{3,5,6,7\},\{4,5,6,7\},\{2,5\},\{2,6\}, \{2,7\}\}$. *The set* $\mathcal{A} = \mathcal{U}^4_8(g^1)$ *is in* $MI^2(8,4,1)$, *and* $E(\mathcal{A}) = [1,7]$. *The set* $g(\mathcal{A})$ *as defined in (3.3) is the following:*

$$\begin{aligned} g(\mathcal{A}) &= \{\{1,2,3,4\},\{1,5,6,7\},\{3,5,6,7\},\{4,5,6,7\}\} \\ &\quad \cup\, \mathcal{U}^3_7(\{\{2,5\},\{2,6\},\{2,7\}\}) \cup \mathcal{U}^4_7(\{\{2,5\},\{2,6\},\{2,7\}\}). \end{aligned}$$

The set $\mathcal{A}' = \mathcal{U}^4_9(g(\mathcal{A}))$ *is not maximal in* $I^2(9,4,1)$, *since, for example, the set* $B = \{2,5,8,9\}$ *intersects every other set in* $\mathcal{A}'$, *but* $B \notin \mathcal{A}'$. *However, there exists a generating set of* $\mathcal{A}$, *namely* g^1, *that has the special property that* $\mathcal{U}^4_v(g) \in I^2(v,4,1)$ *is maximal for all* $v \geq 8$.

We now introduce a new generating set.

Proposition 3.2.7 *Let* $\mathcal{A}$ *be a family in* $MI^s(v,k,t)$, *and* $g(\mathcal{A})$ *be the generating set introduced in (3.3). Denote by* $g'(\mathcal{A})$ *the family given by*

$$g'(\mathcal{A}) = \{G \subseteq \overline{G} : \overline{G} \in g(\mathcal{A}),\ \textit{and}\ \{G\} \cup g(\mathcal{A})\ \textit{is s-wise t-intersecting}\}. \tag{3.5}$$

Then, if $v > 2k-t$, *the family* $g'(\mathcal{A})$ *is a generating set for* $\mathcal{A}$.

Proof. Since $g(\mathcal{A}) \subseteq g'(\mathcal{A})$, we have $\mathcal{A} = \mathcal{U}_v^k(g(\mathcal{A})) \subseteq \mathcal{U}_v^k(g'(\mathcal{A}))$. We have to show $\mathcal{U}_v^k(g'(\mathcal{A})) \subseteq \mathcal{A}$. Let $B \in \mathcal{U}_v^k(g'(\mathcal{A}))$. If $A_1, \ldots, A_{s-1} \in \mathcal{A}$, there exist $G_1, \ldots, G_{s-1} \in g(\mathcal{A})$, such that $A_1 \supseteq G_1, \ldots, A_{s-1} \supseteq G_{s-1}$. Also, $B \supseteq G$, for G such that $\{G\} \cup g(\mathcal{A})$ is s-wise t-intersecting. So $|A_1 \cap \ldots \cap A_{s-1} \cap B| \geq |G_1 \cap \ldots \cap G_{s-1} \cap G| \geq t$. Therefore $\{B\} \cup \mathcal{A}$ is in $I^s(v,k,t)$, and since $\mathcal{A}$ is maximal, we conclude that $B \in \mathcal{A}$. $\Box$

In the next theorem, we show that the generating set $g'(\mathcal{B})$ can be used to produce maximal families for larger v's.

Theorem 3.2.8 *Let $v \geq v_0 \geq 2k - t + 1$, and let $\mathcal{B}$ be a family in $MI^s(v_0,k,t)$. Define $\mathcal{A}$ to be a family in $I^s(v,k,t)$ given by $\mathcal{A} = \mathcal{U}_v^k(g'(\mathcal{B}))$. Then*

1. *$E(\mathcal{B}) = E(\mathcal{A})$;*

2. *the family $\mathcal{A}$ is maximal in $I^s(v,k,t)$; and*

3. *$g'(\mathcal{B}) = g'(\mathcal{A})$.*

Proof. Part 1. Given that $\mathcal{B} \subseteq \mathcal{A}$, we see that $E(\mathcal{B}) \subseteq E(\mathcal{A})$. So, we only need to show that $E(\mathcal{A}) \subseteq E(\mathcal{B})$. Let $e \in E(\mathcal{A})$. Then, there exist $A_1, \ldots, A_s \in \mathcal{A}$ with $e \in A_1 \cap \ldots \cap A_s$ and $|A_1 \cap \ldots \cap A_s| = t$. By construction, $A_i \supseteq G_i$ for some $G_i \in g'(\mathcal{B})$, for $1 \leq i \leq s$. Using the fact that $g'(\mathcal{B})$ is a generating set for $\mathcal{B}$ and Lemma 3.2.2, we get that $|G_1 \cap \ldots \cap G_s| \geq t$. Therefore, we must have $e \in G_1 \cap \ldots \cap G_s \subseteq E(\mathcal{B})$.

Part 2. Clearly $\mathcal{A} \in I^s(v,k,t)$. It is left to show that $\mathcal{A}$ is maximal. Let $C \in [1,v]$, with $|C| = k$ and such that $\mathcal{A} \cup \{C\}$ is s-wise t-intersecting. Considering that $g'(\mathcal{B})$ is a generating set for $\mathcal{A}$, we conclude that $g'(\mathcal{B}) \cup \{C\}$ is a generating set for $\mathcal{A} \cup \{C\}$. Thus, by Lemma 3.2.2, $g'(\mathcal{B}) \cup \{C\}$ is s-wise t-intersecting. Given that $\cup_{G \in g'(\mathcal{B})} G = E(\mathcal{B})$, it follows that $g'(\mathcal{B}) \cup \{C \cap E(\mathcal{B})\}$ is s-wise t-intersecting. It is easy to prove that $C \cap E(\mathcal{B}) \in g'(\mathcal{B})$. Indeed, let B be any k-extension of $C \cap E(\mathcal{B})$ contained in $[1, v_0]$. Thus, $\mathcal{B} \cup \{B\}$ is s-wise t-intersecting, and since $\mathcal{B}$ is maximal, we have $B \in \mathcal{B}$. Considering that $C \cap E(\mathcal{B}) \subseteq B \cap E(\mathcal{B})$ and that $g'(\mathcal{B}) \cup \{C \cap E(\mathcal{B})\}$ is s-wise t-intersecting we conclude that $C \cap E(\mathcal{B}) \in g'(\mathcal{B})$. Therefore, any k-extension of $C \cap E(\mathcal{B})$ contained in $[1,v]$ belongs to $\mathcal{A}$, by definition; in particular, the set $C \in \mathcal{A}$.

Part 3. First, we show that $g'(\mathcal{B}) \subseteq g'(\mathcal{A})$. Let $G \in g'(\mathcal{B})$. Let A be any k-extension of G contained in $[1, v]$; since $\mathcal{A}$ is maximal, it follows that $A \in \mathcal{A}$. Observing that $G \subseteq E(\mathcal{B}) = E(\mathcal{A})$, we conclude that $G \subseteq A \cap E(\mathcal{A})$. Considering that $\{G\} \cup g(\mathcal{A})$ is a generating set for $\mathcal{A}$, by Lemma 3.2.2, $\{G\} \cup g(\mathcal{A})$ must be s-wise t-intersecting. Thus, by the definition of $g'(\mathcal{A})$, we conclude that $G \in g'(\mathcal{A})$.

Finally, we show that $g'(\mathcal{A}) \subseteq g'(\mathcal{B})$. Let $G' \in g'(\mathcal{A})$. Then, by definition, $G' \subseteq A \cap E(\mathcal{A})$, for some $A \in \mathcal{A}$, and $\{G'\} \cup g(\mathcal{A})$ is s-wise t-intersecting. Considering that $\mathcal{B} \subseteq \mathcal{A}$, we see that $g(\mathcal{B}) \subseteq g(\mathcal{A})$. This implies that $\{G'\} \cup g(\mathcal{B})$ is s-wise t-intersecting. We just have to show that $G' \subseteq B \cap g(\mathcal{B})$ for some $B \in \mathcal{B}$. Let B be any k-extension of $A \cap E(\mathcal{A})$ contained in $[1, v_0]$. As $G' \subseteq B$, we know that $B \in \mathcal{A}$. This implies B is a k-extension of some $G \in g'(\mathcal{B})$, by construction of $\mathcal{A}$. In addition, $B \subseteq [1, v_0]$, which implies $B \in \mathcal{B}$. We conclude the proof by observing that $G' = G' \cap E(\mathcal{B}) \subseteq B \cap E(\mathcal{B})$. $\square$

We show that the generating set $g'(\mathcal{A})$ for a family $\mathcal{A}$ is the same as a generating set $g'(\mathcal{B})$ for some family $\mathcal{B}$ with smaller v, provided that v is large enough.

Theorem 3.2.9 *Let $v_0 \geq \max\{2k-t+1, n(k, s, t)\}$, and let $\mathcal{A}$ be a family in $MI^s(v, k, t)$, with $v \geq v_0$. Assume w.l.o.g. that $E(\mathcal{A}) = [1, |E(\mathcal{A})|]$.*

Let $\mathcal{B} = \mathcal{U}^k_{v_0}(g'(\mathcal{A}))$. Then

1. *$E(\mathcal{B}) = E(\mathcal{A})$;*
2. *the family $\mathcal{B}$ is maximal in $I^s(v_0, k, t)$; and*
3. *$g'(\mathcal{B}) = g'(\mathcal{A})$.*

Proof. Part 1. Since $\mathcal{B} \subseteq \mathcal{A}$, it is clear that $E(\mathcal{B}) \subseteq E(\mathcal{A})$. We have to show that $E(\mathcal{A}) \subseteq E(\mathcal{B})$. Let $e \in E(\mathcal{A})$, then there exist sets $A_1, \ldots, A_s \in \mathcal{A}$ with $e \in A_1 \cap \ldots \cap A_s$ and $|A_1 \cap \ldots \cap A_s| = t$. Let $I = A_1 \cap \ldots \cap A_s$. Consider $G_i = A_i \cap E(\mathcal{A})$, for $1 \leq i \leq s$. Then $G_1, \ldots, G_s \in g(\mathcal{A})$, and $I \subseteq G_1 \cap \ldots \cap G_s$. Thus $|G_i| \geq |I| = t$. Moreover, $G_1, \ldots, G_s \in g'(\mathcal{A})$, so $\mathcal{U}^k_{v_0}(\{G_1, \ldots, G_s\}) \subseteq \mathcal{B}$. We choose the following k-extensions of $G_1, \ldots, G_s$ contained in $[1, v_0]$:

- $B_1 = G_1 \cup L_1$, for some $L_1 \in [1, v_0] \setminus (G_1 \cup (G_2 \cap \ldots \cap G_s))$;

- $B_i = G_i \cup L_i$, for some $L_i \subseteq [1, v_0] \setminus (G_i \cup B_1)$, $i = 2, \ldots, s$.

There exists always such an L_1, since $|[1, v_0] \setminus (G_1 \cup (G_2 \cap \ldots \cap G_s))| \geq v_0 - (|G_1| + |G_2 \cap \ldots \cap G_s| - |I|) \geq 2k - t + 1 - (|G_1| + k - t) \geq k - |G_1|$. For $2 \leq i \leq s$, there exists always such an L_i, since $|[1, v_0] \setminus (G_i \cup B_1)| = v_0 - (|G_i| + |B_1| - |G_i \cap B_1|) \geq 2k - t + 1 - |G_i| - k + t \geq k - |G_i|$.

Therefore $B_1, \ldots, B_s \in \mathcal{B}$ and

$$\begin{aligned} B_1 \cap \ldots \cap B_s &= (G_1 \cup L_1) \cap (G_2 \cup L_2) \cap \ldots \cap (G_s \cup L_s) \\ &= (G_1 \cup L_1) \cap G_2 \ldots \cap G_s \qquad \text{(by the choice of } L_2, \ldots, L_s) \\ &= (G_1 \cap G_2 \cap \ldots \cap G_s) \cup (L_1 \cap G_2 \cap \ldots \cap G_s) \\ &= I. \end{aligned}$$

So $e \in B_1 \cap \ldots \cap B_s$ and $|B_1 \cap \ldots \cap B_s| = |I| = t$. Therefore e is essential for $\mathcal{B}$.

Part 2. One can easily check from $\mathcal{B} = \mathcal{U}_{v_0}^k(g'(\mathcal{A}))$ and $\cup_{G \in g'(\mathcal{A})} G \subseteq [1, |E(\mathcal{A})|] \subseteq [1, v_0]$, that $\mathcal{B} \in I^s(v_0, k, t)$. We only have to prove that $\mathcal{B}$ is maximal. Let $C \in [1, v_0]$ such that $\mathcal{B}' = \mathcal{B} \cup \{C\}$ is s-wise t-intersecting. Then, $\overline{g} = g'(\mathcal{A}) \cup \{C\}$ is a generating set of $\mathcal{B}'$. By Lemma 3.2.2, the set $\overline{g}$ is s-wise t-intersecting, which implies that for any $G_1, \ldots, G_{s-1} \in g'(\mathcal{A})$, we have $|G_1 \cap \ldots \cap G_{s-1} \cap C| \geq t$. So, $\mathcal{A} \cup \{C\}$ is s-wise t-intersecting, and since $\mathcal{A}$ is maximal we must have $C \in \mathcal{A}$. Therefore, $C \cap E(\mathcal{A}) \in g'(\mathcal{A})$, which implies $C \in \mathcal{B}$.

Part 3. Let us first prove that $g'(\mathcal{A}) \subseteq g'(\mathcal{B})$. Let $\overline{G} \in g'(\mathcal{A})$. Then, $\overline{G} \subseteq A \cap E(\mathcal{A})$, for some $A \in \mathcal{A}$, and $g(\mathcal{A}) \cup \{\overline{G}\}$ is s-wise t-intersecting. Since $\mathcal{B} \subseteq \mathcal{A}$ and $E(\mathcal{B}) = E(\mathcal{A})$, it follows that $g(\mathcal{B}) \subseteq g(\mathcal{A})$. Thus, $g(\mathcal{B}) \cup \{\overline{G}\}$ is s-wise t-intersecting. In addition, $B = \overline{G} \cup L \in \mathcal{B}$, for any $L \subseteq [1, v_0] \setminus \overline{G}$. Then, $B \cap E(\mathcal{B}) = (\overline{G} \cup L) \cap E(\mathcal{A}) = \overline{G} \cup (L \cap E(\mathcal{A}))$. So, $\overline{G} \subseteq B \cap E(\mathcal{B})$ for some $B \in \mathcal{B}$, and $g(\mathcal{B}) \cup \{\overline{G}\}$ is s-wise t-intersecting. Therefore, $\overline{G} \in g'(\mathcal{B})$.

We must show that $g'(\mathcal{B}) \subseteq g'(\mathcal{A})$. Let $\overline{G} \in g'(\mathcal{B})$. Thus, $\overline{G} \subseteq B \cap E(\mathcal{B})$, for some $B \in \mathcal{B}$, and $g(\mathcal{B}) \cup \{\overline{G}\}$ is s-wise t-intersecting. Given that $\mathcal{B} \subseteq \mathcal{A}$ and $E(\mathcal{B}) = E(\mathcal{A})$, we conclude that $\overline{G} \subseteq B \cap E(\mathcal{B}) = B \cap E(\mathcal{A})$, with $B \in \mathcal{A}$. Let $G_1, \ldots, G_{s-1} \in g(\mathcal{A})$. Considering that $g(\mathcal{A}) \subseteq g'(\mathcal{A}) \subseteq g'(\mathcal{B})$, we see that $G_1, \ldots, G_{s-1} \in g'(\mathcal{B})$. Using the fact that $g'(\mathcal{B})$ is a generating set of $\mathcal{B}$ and $v_0 > 2k - t$, by Lemma 3.2.2, we conclude that

$g'(\mathcal{B})$ is s-wise t-intersecting, and in particular that $|\overline{G} \cap G_1 \cap \ldots \cap G_{s-1}| \geq t$. Therefore, $g(\mathcal{A}) \cup \{\overline{G}\}$ is s-wise t-intersecting, which implies $\overline{G} \in g'(A)$. $\square$

The following corollary implies that the solution of Problem 2 for k, s, t and $v \geq n_0(k,s,t)$ can be obtained from the solution of Problem 2 for k, s, t and $v = n_0(k,s,t)$.

Corollary 3.2.10

Let $G(v) = \{g'(\mathcal{A}) : \mathcal{A} \in MI^s(v,k,t)\}$ and $n_0(k,s,t) = \max\{n(k,s,t), 2k-t+1\}$. If $v_1, v_2 \geq n_0(k,s,t)$, then $G(v_1)$ is isomorphism-equivalent to $G(v_2)$.

Proof. Assume w.l.o.g. that $v_2 \geq v_1$. Theorem 3.2.9 establishes that for every family $\mathcal{A} \in MI^s(v_2,k,t)$, there exists a family $\mathcal{B} \in MI^s(v_1,k,t)$ such that $g'(\mathcal{B}) \sim g'(\mathcal{A})$. Theorem 3.2.8 shows that for every family $\mathcal{B} \in MI^s(v_1,k,t)$, there exists a family $\mathcal{A} \in MI^s(v_2,k,t)$ such that $g'(\mathcal{A}) = g'(\mathcal{B})$. $\square$

Remark 3.2.11 *For $\mathcal{A} \in MI^s(v,k,t)$, the generating set $g'(\mathcal{A})$ is not necessarily minimal with respect to set inclusion. For representing $\mathcal{A}$, we may also use the following generating set:*

$$g''(\mathcal{A}) = \{G \in g'(\mathcal{A}) : \text{ for any } G_1 \in g'(\mathcal{A}), G_1 \subseteq G \text{ implies } G_1 = G\}.$$

It is easy to check that $g''(\mathcal{A})$ is a generating set of $\mathcal{A}$, and that $g''(\mathcal{A})$ satisfies the same properties as $g'(\mathcal{A})$, given in Theorem 3.2.9 and Corollary 3.2.10.

Example 3.2.12 *Let $\mathcal{A} \in MI^2(8,4,1)$ as given in example 3.2.6, with*

$$\begin{aligned} g(\mathcal{A}) &= \{\{1,2,3,4\},\{1,5,6,7\},\{3,5,6,7\},\{4,5,6,7\}\} \\ &\quad \cup\, \mathcal{U}_7^3(\{\{2,5\},\{2,6\},\{2,7\}\}) \cup \mathcal{U}_7^4(\{\{2,5\},\{2,6\},\{2,7\}\}). \end{aligned}$$

Then,

$$\begin{aligned} g'(\mathcal{A}) &= g(\mathcal{A}) \cup \{\{2,5\},\{2,6\},\{2,7\}\}, \text{ and} \\ g''(\mathcal{A}) &= \{\{1,2,3,4\},\{1,5,6,7\},\{3,5,6,7\},\{4,5,6,7\},\{2,5\},\{2,6\},\{2,7\}\}. \end{aligned}$$

3.3 A construction of families in $MI^s(v+1,k+1,t+1)$ using $MI^s(v,k,t)$

In this section, we show how a family in $MI^s(v,k,t)$ can be used to construct a family in $MI^s(v+1,k+1,t+1)$, for any s, v, k and t. We also show that if the construction is applied to two nonisomorphic families, then the resulting families are also nonisomorphic (Theorem 3.3.5). In addition, for $v > 2k-t$, this construction increases the number of essential numbers by exactly one (Proposition 3.3.8).

The idea behind the construction is as follows. Let $\mathcal{A} \in MI^s(v,k,t)$. The family $\mathcal{A}_{(v+1)}$ obtained by adding the element $v+1$ to every set in $\mathcal{A}$ is obviously a family in $I^s(v+1,k+1,t+1)$, but not necessarily maximal. The construction adds appropriate sets to $\mathcal{A}_{(v+1)}$ in order to obtain a family $\mathcal{B}$ that is maximal. In fact, we show that $\mathcal{B}$ is the only maximal family in $MI^s(v+1,k+1,t+1)$ that contains $\mathcal{A}_{(v+1)}$.

Construction 3.3.1 *Let $\mathcal{A}$ be a family in $MI^s(v,k,t)$. Let $\mathcal{A}_{(v+1)} = \{A \cup \{v+1\} : A \in \mathcal{A}\}$. For any $A \in \mathcal{A}$, denote $\mathcal{E}(A) = \{(C_1,\dots,C_{s-1}) \in \mathcal{A} : |C_1 \cap \dots \cap C_{s-1} \cap A| = t\}$, and $\mathcal{L}(A) = \Big(\cap_{(C_1,\dots,C_{s-1})\in\mathcal{E}(A)}(C_1 \cap \dots \cap C_{s-1})\Big) \setminus A$. Let, for any $A \in \mathcal{A}$,*

$$\mathcal{A}^*(A) = \begin{cases} \{A \cup \{l\} : l \in \mathcal{L}(A)\} & \text{if } \mathcal{E}(A) \neq \emptyset, \\ \{A \cup \{i\} : i \in [1,v] \setminus A\} & \text{if } \mathcal{E}(A) = \emptyset. \end{cases},$$

and $\mathcal{A}^* = \cup_{A\in\mathcal{A}} \mathcal{A}^*(A)$. Define $\mathcal{B}$ to be the family $\mathcal{A}_{(v+1)} \cup \mathcal{A}^*$.

Theorem 3.3.2 *Let $\mathcal{A}$ be a family in $MI^s(v,k,t)$. Then the family $\mathcal{B} = \mathcal{A}_{(v+1)} \cup \mathcal{A}^*$ given by Construction 3.3.1 is a family in $MI^s(v+1,k+1,t+1)$. Moreover, family $\mathcal{B}$ is the only family in $MI^s(v+1,k+1,t+1)$ that contains the subfamily $\mathcal{A}_{(v+1)}$.*

Proof. First, we will prove that $\mathcal{B} \in I^s(v+1,k+1,t+1)$. It is clear that $\mathcal{B}$ is $(k+1)$-uniform, so it remains to prove that $\mathcal{B}$ is s-wise $(t+1)$-intersecting. Let $B_1,\dots,B_s \in \mathcal{B}$, we need to show that $|B_1 \cap \dots \cap B_s| \geq t+1$. We split the analysis into two cases.

1. If $B_1,\dots,B_s \in \mathcal{A}_{(v+1)}$, then it is trivial that $|B_1 \cap \dots \cap B_s| \geq t+1$.

2. If for some i, $1 \leq i \leq s$, $B_i \in \mathcal{A}^*$, we assume w.l.o.g. that $B_1 \in \mathcal{A}^*$. We know that there exist $A_1, \ldots, A_s \in \mathcal{A}$ such that $B_i \supseteq A_i$, for $1 \leq i \leq s$. If $|A_1 \cap \ldots \cap A_s| \geq t+1$, then we are done. This includes the cases when $\mathcal{E}(A_1) = \emptyset$, for in such case $|A_1 \cap C_1 \cap \ldots \cap C_{s-1}| \geq t+1$, for any $C_1, \ldots, C_{s-1} \in \mathcal{A}$. When $|A_1 \cap \ldots \cap A_s| = t$, we have $\mathcal{E}(A_1) \neq \emptyset$, and by the construction of $\mathcal{A}^*(A_1)$, we must have $B_1 = A_1 \cup \{l_1\}$, with $l_1 \in \mathcal{L}(A_1)$. But, by the construction of $\mathcal{L}(A_1)$, we must have $l_1 \in (A_2 \cap \ldots \cap A_s) \setminus A_1$ then $|B_1 \cap \ldots \cap B_s| \geq |A_1 \cap \ldots \cap A_s| + 1 = t+1$.

We must show that $\mathcal{B}$ is a maximal family in $I(v+1, k+1, t+1)$, and $\mathcal{B}$ is the only maximal family in $I(v+1, k+1, t+1)$ that contains the subfamily $\mathcal{A}_{(v+1)}$. Let $S \subseteq [1, v+1]$ be an arbitrary $(k+1)$-set that is s-wise $(t+1)$-intersecting with any $s-1$ sets $B_1, \ldots, B_{s-1} \in \mathcal{A}_{(v+1)}$. We prove both of the above statements by showing that $S \in \mathcal{B}$. If $v+1 \in S$, then $S \setminus \{v+1\}$ must be s-wise t-intersecting with any $s-1$ sets $A_1, \ldots, A_{s-1} \in \mathcal{A}$. Since $\mathcal{A}$ is maximal, $S \setminus \{v+1\} \in \mathcal{A}$, and so $S \in \mathcal{B}$. If $v+1 \notin S$, then let $a \in S$ and call $S_a = S \setminus \{a\}$. Thus $\{S\} \cup \mathcal{A}$ is s-wise $(t+1)$-intersecting, which implies that $\{S_a\} \cup \mathcal{A}$ is s-wise t-intersecting, and then $S_a \in \mathcal{A}$. If $\mathcal{E}(S_a) = \emptyset$, by construction of $\mathcal{A}^*(S_a)$, we have $S = S_a \cup \{a\} \in \mathcal{B}$. Otherwise, if $\mathcal{E}(S_a) \neq \emptyset$, for any $(C_1, \ldots, C_{s-1}) \in \mathcal{E}(S_a)$, we have $|C_1 \cap \ldots \cap C_{s-1} \cap S_a| = t$, but $|C_1 \cap \ldots \cap C_{s-1} \cap S| \geq t+1$, and so we must have $a \in C_1 \cap \ldots \cap C_{s-1}$. Therefore $a \in \mathcal{L}(S_a)$, and then $S = S_a \cup \{a\} \in \mathcal{B}$. $\square$

Example 3.3.3 *Let $\mathcal{A} \in MI^2(6,3,1)$ given by the first column of the following table. The set $\mathcal{B} = \mathcal{A}_{(7)} \cup \mathcal{A}^*(\mathcal{A})$ is in $MI^2(7,4,2)$.*

We introduce some notation that will be used in the next theorem, as well as in the sections to follow.

Notation 3.3.4 *For any family $\mathcal{B}$ of subsets of $[1, v]$, and for any $x, y, z \in [1, v]$, we denote:*

$$\begin{aligned} \mathcal{B}_x &= \{B \in \mathcal{B} : x \in B\}, & \mathcal{B}_{a,b} &= \mathcal{B}_a \cap \mathcal{B}_b, \quad \text{for } a, b \in \{x, \overline{x}, y, \overline{y}\}, \\ \mathcal{B}_{\overline{x}} &= \mathcal{B} \setminus \mathcal{B}_x, & \mathcal{B}_{a,b,c} &= \mathcal{B}_a \cap \mathcal{B}_b \cap \mathcal{B}_c, \quad \text{for } a, b, c \in \{x, \overline{x}, y, \overline{y}, z, \overline{z}\}. \end{aligned}$$

A	$\mathcal{E}(A)$	$\mathcal{L}(A)$	$\mathcal{A}^*(A)$	$\mathcal{A}_{(7)}$
{1,2,3}	Ø		{1,2,3,**4**} {1,2,3,**5**} {1,2,3,**6**}	{1,2,3,**7**}
{1,2,4}	{1,**3**,5} {1,**3**,6} {2,**3**,5} {2,**3**,6}	{**3**}	{1,2,**3**,4}	{1,2,4,**7**}
{1,2,5}	{1,**3**,4} {1,**3**,6} {2,**3**,4} {2,**3**,6}	{**3**}	{1,2,**3**,5}	{1,2,5,**7**}
{1,2,6}	{1,**3**,4} {1,**3**,5} {2,**3**,4} {2,**3**,5}	{**3**}	{1,2,**3**,6}	{1,2,6,**7**}
{1,3,4}	{1,**2**,5} {1,**2**,6} {**2**,3,5} {**2**,3,6}	{**2**}	{1,**2**,3,4}	{1,3,4,**7**}
{1,3,5}	{1,**2**,4} {1,**2**,6} {**2**,3,4} {**2**,3,6}	{**2**}	{1,**2**,3,5}	{1,3,5,**7**}
{1,3,6}	{1,**2**,4} {1,**2**,5} {**2**,3,4} {**2**,3,5}	{**2**}	{1,**2**,3,6}	{1,3,6,**7**}
{2,3,4}	{**1**,2,5} {**1**,2,6} {**1**,3,5} {**1**,3,6}	{**1**}	{**1**,2,3,4}	{2,3,4,**7**}
{2,3,5}	{**1**,2,4} {**1**,2,6} {**1**,3,4} {**1**,3,6}	{**1**}	{**1**,2,3,5}	{2,3,5,**7**}
{2,3,6}	{**1**,2,4} {**1**,2,5} {**1**,3,4} {**1**,3,5}	{**1**}	{**1**,2,3,6}	{2,3,6,**7**}

The following theorem shows that if Construction 3.3.1 is applied to nonisomorphic families, then the resulting families are also nonisomorphic.

Theorem 3.3.5 *Let $\mathcal{A}^1, \mathcal{A}^2 \in MI^s(v,k,t)$, and let $\mathcal{B}^1$ and $\mathcal{B}^2$ be the families in $MI^s(v+1,k+1,t+1)$ given by the application of Construction 3.3.1 to the families $\mathcal{A}^1$ and $\mathcal{A}^2$, respectively. Then $\mathcal{B}^1 \sim \mathcal{B}^2$ implies $\mathcal{A}^1 \sim \mathcal{A}^2$.*

Proof. Let π be the permutation on $[1, v+1]$ such that $\mathcal{B}^2 = \pi(\mathcal{B}^1)$. If $\pi(v+1) = v+1$, then trivially $\mathcal{A}^2 = \pi(\mathcal{A}^1)$. Otherwise, we can assume w.l.o.g. that $\pi(v+1) = v$. Let us call $\mathcal{B} = \mathcal{B}^2 = \pi(\mathcal{B}^1)$. For a family $\mathcal{C}$ we denote $\mathcal{C} \setminus \{x\} = \{C \setminus \{x\} : C \in \mathcal{C}\}$; we can then write:

$$\begin{aligned} \pi(\mathcal{A}^1) &= \mathcal{B}_{v,v+1} \setminus \{v\} \cup \mathcal{B}_{v,\overline{v+1}} \setminus \{v\} \\ \mathcal{A}^2 &= \mathcal{B}_{v,v+1} \setminus \{v+1\} \cup \mathcal{B}_{\overline{v},v+1} \setminus \{v+1\} \end{aligned}$$

We want to show $\pi(\mathcal{A}^1) \sim \mathcal{A}^2$. It is clear that $\mathcal{B}_{v,v+1} \setminus \{v\} \sim \mathcal{B}_{v,v+1} \setminus \{v+1\}$. It is enough to prove that $\mathcal{B}_{v,\overline{v+1}} \setminus \{v\} = \mathcal{B}_{\overline{v},v+1} \setminus \{v+1\}$.

We claim that for any $\overline{A} \in \mathcal{B}_{v,\overline{v+1}} \setminus \{v\}$, the set $\left\{\overline{A}\right\} \cup \mathcal{A}^2$ is s-wise t-intersecting. By definition, $\mathcal{A}^2$ is s-wise t-intersecting. Let $A_1, \ldots, A_{s-1} \in \mathcal{A}^2$ and let $B_i = A_i \cup \{v+1\}$, for $1 \le i \le s-1$, $\overline{B} = \overline{A} \cup \{v\}$. Observe that $B_1, \ldots, B_{s-1}, \overline{B} \in B$. Let us call $I_A = A_1 \cap \ldots \cap A_{s-1} \cap \overline{A}$ and $I_B = B_1 \cap \ldots \cap B_{s-1} \cap \overline{B}$. First, we observe that $I_A = I_B \setminus \{v, v+1\}$, and since $\overline{B} \in \mathcal{B}_{v,\overline{v+1}}$, we know that $v+1 \notin I_B$. So $|I_A| = |I_B \setminus \{v\}| \ge (t+1)-1 = t$. So, $\left\{\overline{A}\right\} \cup \mathcal{A}^2$ is s-wise t-intersecting.

Then, since $\mathcal{A}^2$ is maximal it follows that $\overline{A} \in \mathcal{A}^2$. Therefore, $\mathcal{B}_{v,\overline{v+1}} \setminus \{v\} \subseteq \mathcal{A}^2$, which means $\mathcal{B}_{v,\overline{v+1}} \setminus \{v\} \subseteq \mathcal{B}_{\overline{v},v+1} \setminus \{v+1\}$. By an analogous argument, we show that $\mathcal{B}_{\overline{v},v+1} \setminus \{v+1\} \subseteq \mathcal{B}_{v,\overline{v+1}} \setminus \{v\}$. Therefore, $\mathcal{B}_{\overline{v},v+1} \setminus \{v+1\} = \mathcal{B}_{v,\overline{v+1}} \setminus \{v\}$, which completes the proof. $\square$

An immediate consequence of the previous theorem is that, for any $s \ge 2$, the number of nonisomorphic families is nondecreasing, when parameters v, k and t are increased by one. Recall the meaning of $D(\cdot)$ from Notation 3.1.2.

Corollary 3.3.6 *For any s, v, k and $t \ge 1$, we have*

$$D(MI^s(v+1, k+1, t+1)) \ge D(MI^s(v, k, t)).$$

The following lemma is a well known result for $s = 2$ (see, e.g., [33]), saying basically that, for $v \ge 2k - t$, if $\mathcal{A} \in MI^s(v,k,t)$, then $\mathcal{A} \notin MI^s(v,k,t+1)$.

Lemma 3.3.7 *Let $v \ge 2k - t$ and $\mathcal{A} \in MI^s(v,k,t)$. Then $E(\mathcal{A}) \ne \emptyset$.*

Proof. Suppose to the contrary that $E(\mathcal{A}) = \emptyset$. Then, for every s sets $A_1, \ldots, A_s \in \mathcal{A}$, we have $|A_1 \cap \ldots \cap A_s| \ge t+1$. Let $B_1, \ldots, B_s \in \mathcal{A}$ such that $q = |B_1 \cap \ldots \cap B_s|$ is minimum. Let $I = B_1 \cap \ldots \cap B_{s-1}$. We know that $q = |I \cap B_s| \ge t+1$. Let $a \in I \cap B_s$, and take $\overline{B} = (B_s \setminus \{a\}) \cup \{l\}$ for some $l \in [1,v] \setminus (B_s \cup I)$. There exist such an l, since $v - |B_s \cup I| \ge 2k - t - (2k - (t+1)) = 1$. We claim $\overline{B}$ is s-wise t-intersecting with any other $s-1$ sets $A_1, \ldots, A_{s-1} \in \mathcal{A}$, since B is $(t+1)$-intersecting with $A_1, \ldots, A_{s-1}$. Since $\mathcal{A}$ is maximal, then $\overline{B} \in \mathcal{A}$. But $|B_1 \cap \ldots \cap B_{s-1} \cap \overline{B}| = |I \cap \overline{B}| = |I \cap B_s| - 1 = q - 1$, which contradicts the fact that q is a minimum intersection size among any s sets in $\mathcal{A}$. $\square$

We show that when we apply the previous construction for $v \ge 2k - t$, the number of essential numbers increases by exactly one.

Proposition 3.3.8 *Let $v \geq 2k - t$. Let $\mathcal{A}$ be a family in $MI^s(v, k, t)$, and let the family $\mathcal{B} \in MI^s(v+1, k+1, t+1)$ be the family given by Construction 3.3.1. Then $E(\mathcal{B}) = E(\mathcal{A}) \cup \{v+1\}$.*

Proof. It is easy to prove that $E(\mathcal{A}) \cup \{v+1\} \subseteq E(\mathcal{B})$. Indeed, for any sets $S_1, \ldots, S_s \in \mathcal{A}$, with $|S_1 \cap \ldots \cap S_s| = t$, the sets $(S_1 \cup \{v+1\}), \ldots, (S_s \cup \{v+1\}) \in \mathcal{B}$, have $|(S_1 \cup \{v+1\}) \cap \ldots \cap (S_s \cup \{v+1\})| = t+1$. This shows $E(\mathcal{A}) \subseteq E(\mathcal{B})$. By Lemma 3.3.7, there exist such sets, and so $v+1$ is essential for $\mathcal{B}$.

We prove that $E(\mathcal{B}) \subseteq E(\mathcal{A}) \cup \{v+1\}$. Let $e \in E(\mathcal{B})$, $e \neq v+1$. We must show $e \in E(\mathcal{A})$. Since $e \in E(\mathcal{B})$, there exist $B_1, \ldots, B_s \in \mathcal{B}$ with $e \in B_1 \cap \ldots \cap B_s$, and $|B_1 \cap \ldots \cap B_s| = t+1 \geq 2$.

If $v+1 \in B_1 \cap \ldots \cap B_s$, then the sets $B_i \in \mathcal{A}_{(v+1)}$ for all $1 \leq i \leq s$. Let $C_i = B_i \setminus \{v+1\}$, for $1 \leq i \leq s$; then, $C_1, \ldots C_s \in \mathcal{A}$. Therefore, we have $C_1, \ldots, C_s \in \mathcal{A}$, with $e \in C_1 \cap \ldots \cap C_s$ and $|C_1 \cap \ldots \cap C_s| = |B_1 \cap \ldots \cap B_s| - 1 = t$, which implies $e \in E(\mathcal{A})$.

If $v+1 \notin B_1 \cap \ldots \cap B_s$, then let $f \in B_1 \cap \ldots \cap B_s$, such that $f \neq e$, and define the sets C_i for $1 \leq i \leq s$, as follows:

$$C_i = \begin{cases} B_i \setminus \{v+1\}, & \text{if } v+1 \in B_i, \\ B_i \setminus \{f\}, & \text{otherwise.} \end{cases}$$

We claim that $C_i \in \mathcal{A}$, for all $1 \leq i \leq s$. Indeed, if $v+1 \in B_i$ then $B_i \in \mathcal{A}_{(v+1)}$, and so $C_i = B_i \setminus \{v+1\} \in \mathcal{A}$. Otherwise, if $v+1 \notin B_i$, the fact that $\{\mathcal{B}_i\} \cup \mathcal{A}_{(v+1)}$ is s-wise $(t+1)$-intersecting implies $\{B_i\} \cup \mathcal{A}$ is s-wise $(t+1)$-intersecting. This implies that $\{C_i = B_i \setminus \{f\}\} \cup \mathcal{A}$ is s-wise t-intersecting, and since $\mathcal{A}$ is maximal, we conclude that $C_i \in \mathcal{A}$. We know that $|C_1 \cap \ldots \cap C_s| = |(B_1 \cap \ldots \cap B_s) \setminus \{f\}| = t$, and $e \in C_1 \cap \ldots \cap C_s$, which implies $e \in E(\mathcal{A})$. □

3.4 All maximal k-uniform pairwise t-intersecting families for $k \leq t+2$

In this section, we solve Problem 2 for the case of $s = 2$ and $k = t+1, t+2$. The case of $k = t+1$ is simple, but we include it here for the sake of completeness.

3.4.1 Determination of all families in $MI^2(v, t+1, t)$

In this subsection, we determine all families $\mathcal{A} \in MI^2(v, t+1, t)$ for arbitrary $v \geq t+2$, and show that $n(t+1, 2, t) = t+2$, for all $t \geq 1$.

We will use the following notation, for any k and v: $\binom{[v]}{k} = \{B \subseteq [1, v] : |B| = k\}$.

Proposition 3.4.1 *Let $t \geq 1$ and $v \geq t+2$, and $\mathcal{A} \in MI^2(v, t+1, t)$. The following hold:*

1. *if $v = t+2$, then $\mathcal{A} = \binom{[t+2]}{t+1}$;*
2. *if $v \geq t+3$, then either $\mathcal{A} \sim \mathcal{B}' = \{B \subseteq \binom{[v]}{t+1} : B \supseteq [1, t]\}$ or $\mathcal{A} \sim \binom{[t+2]}{t+1}$.*

Proof. Part 1. The set $\binom{[t+2]}{t+1}$ containing every possible subset is t-intersecting.
Part 2. Let $\mathcal{A} \in MI^2(v, t+1, t)$, with $v \geq t+3$. First, we observe that any distinct sets $A_1, A_2 \in \mathcal{A}$ satisfy $|A_1 \cap A_2| = t$. We analyze the following two cases.
Case a: Suppose that there exist distinct sets $A_1, A_2, A_3 \in \mathcal{A}$ with $|A_1 \cap A_2 \cap A_3| = t$. Assume w.l.o.g. that $A_1 = [1, t] \cup \{t+1\}$, $A_2 = [1, t] \cup \{t+2\}$ and $A_3 = [1, t] \cup \{t+3\}$. It is easy to conclude that any $A \in \mathcal{A}$ that is pairwise t-intersecting with A_1, A_2 and A_3 must contain $[1, t]$, so $\mathcal{A} \subseteq \mathcal{B}'$. Since $\mathcal{A}$ is maximal then $\mathcal{A} = \mathcal{B}'$.
Case b: Suppose that $|A_1 \cap A_2 \cap A_3| < t$, for all distinct sets $A_1, A_2, A_3 \in \mathcal{A}$. We can assume w.l.o.g. that $A = [1, t] \cup \{t+1\}$, $B = [1, t] \cup \{t+2\} \in \mathcal{A}$. For every $C \in \mathcal{A}$, $C \neq A, B$, we must have $|C \cap [1, t]| = t-1$ and $C \supseteq \{t+1, t+2\}$. Therefore, $\mathcal{A} \subseteq \binom{[t+2]}{t+1}$, and since $\mathcal{A}$ is maximal, then $\mathcal{A} = \binom{[t+2]}{t+1}$. $\square$

Remark 3.4.2 *The generating sets $g'(\mathcal{A})$ for any set $\mathcal{A} \in MI^2(v, t+1, t)$ and $v \geq t+3$, are isomorphic to one of the families:*

G_1	G_2
$\{1, \ldots, t\}$	$\{1, \ldots, t+2\} \setminus \{1\}$ $\{1, \ldots, t+2\} \setminus \{2\}$ $\vdots$ $\{1, \ldots, t+2\} \setminus \{t+2\}$

Corollary 3.4.3 *For $t \geq 1$, we have $n(t+1,2,t) = t+2$.*

Proof. From Corollary 3.2.5, we have $n(t+1,2,t) = E_{max}(t+1,2,t)$, and by Proposition 3.4.1 we get $E_{max}(t+1,2,t) = t+2$. □

3.4.2 Determination of all families in $MI^2(v,t+2,t)$

In this subsection, we show that Construction 3.3.1 is enough to generate all the sets in $MI^2(v+1,t+3,t+1)$ when applied to all the sets in $MI^2(v,t+2,t)$, for any v and $t \geq 2$ (Theorem 3.4.6). We also show that $n(t+2,2,t) = t+6$ (Theorem 3.4.8). Using these two results combined with results from previous sections, we manage to reduce Problem 2 (for any $s=2$, v, t and $k=t+2$) to the solution of Problem 2 for $(s,v,k,t) \in \{(2,7,3,1),(2,8,4,2)\}$ (Theorem 3.4.9). This solution is obtained by computer and shown in Table 3.1.

Throughout this section, we will largely use Notation 3.3.4. In addition, for any $\mathcal{A} \in I^s(v,k,t)$, and $a \in [1,v]$, we denote by n_a the number of sets in $\mathcal{A}$ containing a, that is $n_a = |\mathcal{A}_a|$.

Lemma 3.4.4 *Let $\mathcal{A}$ be a family in $MI^2(v,t+2,t)$, for $t \geq 3$. Let $x \in [1,v]$ such that $n_x = \max_{i\in[1,v]} n_i$. Then the family $\mathcal{A}_{\overline{x}}$ is pairwise $(t+1)$-intersecting.*

Proof. It is clear that $\mathcal{A}_{\overline{x}}$ is pairwise t-intersecting, since $\mathcal{A}_{\overline{x}} \subseteq \mathcal{A}$. In order to prove that $\mathcal{A}_{\overline{x}}$ is, in addition, pairwise $(t+1)$-intersecting, we suppose to the contrary that there exist $A_1, A_2 \in \mathcal{A}_{\overline{x}}$, such that $|A_1 \cap A_2| = t$. Then A_1 and A_2 must be of the form

$$A_1 = \{y_1, y_2, \ldots, y_t, \alpha_1, \beta_1\}, \text{ and } A_2 = \{y_1, y_2, \ldots, y_t, \alpha_2, \beta_2\},$$

for distinct $y_1, y_2, \ldots, y_t, \alpha_1, \beta_1, \alpha_2, \beta_2$, different from x.

For any i, since $n_{y_i} \leq n_x$ we must have

$$|\mathcal{A}_{\overline{y_i},x}| \geq |\mathcal{A}_{y_i,\overline{x}}| \geq 2. \tag{3.6}$$

We claim that $\mathcal{A}_{\overline{y_i},x}$ cannot be pairwise $(t+1)$-intersecting. Indeed, if it was so, it would imply that the family obtained by replacing x by y_i in $\mathcal{A}_{\overline{y_i},x}$, denoted by $\mathcal{A}_{\overline{y_i},x} -$

$\{x\} + \{y_i\}$, would t-intersect all other sets in $\mathcal{A}$. Then, since $\mathcal{A}$ is maximal, we would have $\mathcal{A}_{\overline{y_i},x} - \{x\} + \{y_i\} \subseteq \mathcal{A}$. Consequently, we would get $|\mathcal{A}_{y_i,\overline{x}}| \geq |(\mathcal{A}_{\overline{y_i},x} - \{x\} + \{y_i\}) \cup \{A_1\} \cup \{A_2\}| = |\mathcal{A}_{\overline{y_i},x}| + 2 > |\mathcal{A}_{\overline{y_i},x}|$, contradicting (3.6). Therefore, for $i = 1, \ldots, t$,

$$\text{there exist } B_1^i, B_2^i \in \mathcal{A}_{\overline{y_i},x}, \text{with } |B_1^i \cap B_2^i| = t. \tag{3.7}$$

Since $B_1^i, B_2^i \in \mathcal{A}_{\overline{y_i},x}$ and must also t-intersect both A_1 and A_2, they must be of the form

$$B_1^i = \{x, y_1, y_2, \ldots, y_t\} \setminus \{y_i\} \cup Z_1^i, \quad B_2^i = \{x, y_1, y_2, \ldots, y_t\} \setminus \{y_i\} \cup Z_2^i,$$

with $Z_1^i, Z_2^i \in \{\{\alpha_1, \alpha_2\}, \{\alpha_1, \beta_2\}, \{\beta_1, \alpha_2\}, \{\beta_1, \beta_2\}\}$. Condition (3.7) translates into $Z_1^i \cap Z_2^i = \emptyset$, for all $i = 1, \ldots, t$, which implies $\{Z_1^i, Z_2^i\}$ equals either $\{\{\alpha_1, \alpha_2\}, \{\beta_1, \beta_2\}\}$ or $\{\{\alpha_1, \beta_2\}, \{\beta_1, \alpha_2\}\}$. Since $t \geq 3$, by the pigeon-hole principle for some $i_1 \neq i_2$, we have $\{Z_1^{i_1}, Z_2^{i_1}\} = \{Z_1^{i_2}, Z_2^{i_2}\}$. So, w.l.o.g., $Z_1^{i_1} = Z_1^{i_2}$, which implies $|B_1^{i_1} \cap B_2^{i_2}| = t - 1 < t$, i.e., a contradiction. □

Lemma 3.4.5 *Let $\mathcal{A}$ be a family in $MI^2(v, t+2, t)$, for $t \geq 3$. Let $x \in [1, v]$ such that $n_x = \max_{i \in [1,v]} n_i$. Then for any $B \in \mathcal{A}_{\overline{x}}$ and for any $y \in B$ we have $(B \setminus \{y\} \cup \{x\}) \in \mathcal{A}$.*

Proof. Let $B \in \mathcal{A}_{\overline{x}}$ and $y \in B$. Let $A \in \mathcal{A}$. If $A \in \mathcal{A}_x$, then $|A \cap (B \setminus \{y\} \cup \{x\})| \geq |A \cap B| \geq t$. Otherwise, if $A \in \mathcal{A}_{\overline{x}}$, by Lemma 3.4.4, we have $|A \cap B| \geq t + 1$, which implies $|A \cap (B \setminus \{y\} \cup \{x\})| \geq t$. Since $\mathcal{A}$ is maximal, we must have $(B \setminus \{y\} \cup \{x\}) \in \mathcal{A}$. □

The following theorem implies that, for any $t \geq 3$, the sets in $MI^2(v, t+2, t)$ are the ones obtained from maximal sets of $MI^2(v-1, t+1, t-1)$ using Construction 3.3.2.

Theorem 3.4.6 *Let $\mathcal{A}$ be a family in $MI^2(v, t+2, t)$, for $t \geq 3$. Let $x \in [1, v]$ such that $n_x = \max_{i \in [1,v]} n_i$. Then, the family $\mathcal{A}' = \mathcal{A}_x \setminus \{x\}$ is a family in $MI^2(v-1, t+1, t-1)$.*

Proof. By definition, $\mathcal{A}' \in I^2(v-1, t+1, t-1)$. It remains to prove that $\mathcal{A}'$ is maximal in $I^2(v-1, t+1, t-1)$. Let $L' \subseteq [1, v] \setminus \{x\}$, with $|L'| = t+1$, and such that $|L' \cap A| \geq t-1$, for all $A \in \mathcal{A}'$. Denote $L = L' \cup \{x\}$. We must prove that $L' \in \mathcal{A}'$ or equivalently that

$L \in \mathcal{A}$. By the maximality of $\mathcal{A}$, its enough to prove that for any $B \in \mathcal{A}$, we have $|B \cap L| \geq t$.

Let $B \in \mathcal{A}$. If $B \in \mathcal{A}_x$, then $B \setminus \{x\} \in \mathcal{A}'$, which implies $|B \setminus \{x\} \cap L'| = t - 1$ and trivially $|B \cap L| \geq t$. So, assume $B \in \mathcal{A}_{\overline{x}}$. We claim that $B \cap L' \neq \emptyset$. Indeed, let $A' \in \mathcal{A}'$. Then $|A'| = t + 1$, $|A' \cap B| \geq t \geq 3$ and $|A' \cap L'| \geq t - 1$. Then,

$$|B \cap L'| \geq |A' \cap B \cap L'| \geq |A' \cap B| + |A' \cap L'| - |A'| \geq t + (t-1) - (t+1) = t - 2 \geq 1.$$

Let $y \in B \cap L'$. By Lemma 3.4.5, $(B \setminus \{y\} \cup \{x\}) \in \mathcal{A}$ and so $(B \setminus \{y\}) \in \mathcal{A}'$. By definition of L', we have $|(B \setminus \{y\}) \cap L'| \geq t - 1$. Since $y \in B \cap L'$, we get $|B \cap L'| \geq t$, and finally $|B \cap L| \geq t$. $\square$

We examine the consequences of the previous theorem on the size of kernels for $s = 2$ and $t = k - 2 \geq 2$.

Corollary 3.4.7 *Let $t = k - 2 \geq 2$. Then $n(t+2, 2, t) = n(4, 2, 2) + t - 2$.*

Proof. Let $v \geq t + 5$ and let $\mathcal{B} \in MI^2(v, t+2, t)$. By Theorem 3.4.6, there exist $\mathcal{A} \in MI^2(v-(t-2), 4, 2)$ such that each $\mathcal{B}$ is obtained by $t-2$ applications of Construction 3.3.1 to $\mathcal{A}$. Thus, by Proposition 3.3.8, we have $|E(\mathcal{B})| = |E(\mathcal{A})| + t - 2$. Finally, using Corollary 3.2.5, we get $n(t+2, 2, t) = n(4, 2, 2) + t - 2$. $\square$

Finally, we just need to determine $n(3, 2, 1)$ and $n(4, 2, 2)$ in order to obtain $n(t+2, 2, t)$ for any $t \geq 1$.

Theorem 3.4.8 *For any $t \geq 1$, we have $n(t+2, 2, t) = t + 6$.*

Proof. First, we observe that $n(3, 2, 1) \geq 7$ and $n(4, 2, 2) \geq 8$, since families $G_8 \in MI^2(7, 3, 1)$ and $H_8 \in MI^2(8, 4, 2)$ (see Table 3.1) are such that $E(G_8) = 7$ and $E(H_8) = 8$. In addition, if $n(4, 2, 2) \leq 8$, using the fact that $n(k-1, s, t-1) \leq n(k, s, t) - 1$, we get $n(3, 2, 1) \leq 7$. This implies $n(3, 2, 1) = 7$ and $n(4, 2, 2) = 8$, which combined with Corollary 3.4.7, leads to $n(t+2, 2, t) = t + 6$, for $t \geq 1$. Thus, it is enough to show that $n(4, 2, 2) \leq 8$.

Let $\mathcal{A} \in MI^2(v, 4, 2)$ for arbitrary $v \geq 8$, and let $A, B \in \mathcal{A}$ be an essential pair for $\mathcal{A}$ (there exit such a pair by Lemma 3.3.7). We can assume w.l.o.g. that $A = \{1, 2, 3, 4\}$

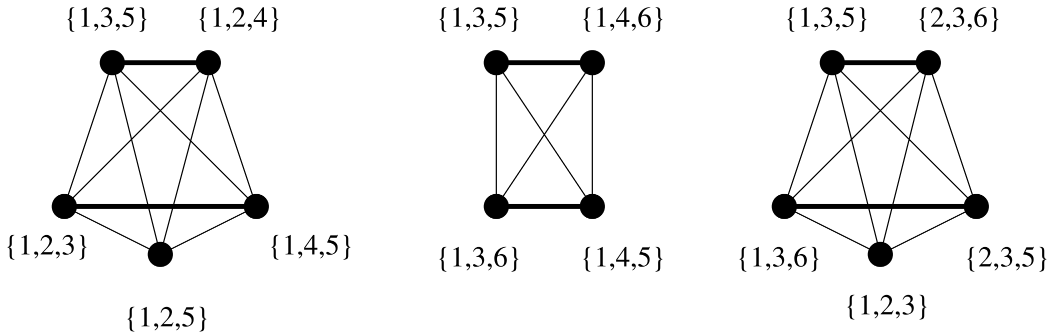

Figure 3.1: Coloured edges of graphs $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$.

and $B = \{1, 2, 5, 6\}$. We can also assume that $\mathcal{A}$ has no set of the form $\{1, 2, a, b\}$ with $a, b \in [1, v] \setminus [1, 6]$, for this would imply that $\{1, 2\}$ must be contained in every other set in $\mathcal{A}$ and then $|E(\mathcal{A})| = 2$. Therefore, the remaining sets that are possibly contained in $\mathcal{A}$ and that include an element $x \notin [1, 6]$ are of the form:

$$\{1, 2, a, x\}, \qquad \text{for } a \in \{3, 4, 5, 6\} \text{ and } x \in [7, v]; \tag{3.8}$$

$$\{b, c, e, x\}, \qquad \text{for } b \in \{1, 2\}, \{c, e\} \subseteq \{\{3, 5\}, \{3, 6\}, \{4, 5\}, \{4, 6\}\}, \tag{3.9}$$
$$\text{and } x \in [7, v].$$

Therefore, any essential element $x \in [7, v]$ must have an essential pair chosen from sets in (3.8) and (3.9). We will show there are at most 2 such essential elements, which implies $|E(\mathcal{A})| \leq 8$.

Define a graph $\mathcal{G}$ with vertices corresponding to the possible triples accompanying an essential element $x \in [7, v]$, given by (3.8) and (3.9), and edges placed whenever the triples corresponding to its endings intersect. Colour the edges whose triples corresponding to its endings intersect in exactly 1 element. For each essential element $x \in [7, v]$ for $\mathcal{A}$, let $e(x)$ be an edge of $\mathcal{G}$ connecting some triples T_1 and T_2 such that $T_1 \cup \{x\}$ and $T_2 \cup \{x\}$ is an essential pair for $\mathcal{A}$. Let $E' = \{e(x) : x \in E(\mathcal{A}) \setminus [1, 6]\}$. Let $\mathcal{G}'$ be the subgraph of $\mathcal{G}$ induced by the set of all vertices that are endings for some edge in E'. Since $\mathcal{A}$ is 2-intersecting, $\mathcal{G}'$ is a complete graph whose coloured edges form a matching. We will show that such an E' must satisfy $|E'| \leq 2$, which implies $|E(\mathcal{A})| \leq 8$.

We reduce the size of the subgraphs we have to look at, by observing that another

Generating sets for all nonisomorphic families in $MI^2(v,3,1)$, for $v \geq 7$	
i	G^i
1	{1}
2	{1,2} {1,3} {2,3}
3	{1,2} {1,3} {1,4} {2,3,4}
4	{1,2} {1,3} {1,4,5} {2,3,4} {2,3,5}
5	{1,2} {1,3,4} {1,3,5} {1,4,5} {2,3,4} {2,3,5} {2,4,5}
6	{1,2} {1,3,4} {1,3,6} {1,4,5} {1,5,6} {2,3,5} {2,4,6}
7	{1,2} {1,3,4} {1,3,6} {1,4,5} {2,3,4} {2,3,5} {2,4,6}
8	{1,2,3} {1,4,5} {1,6,7} {2,4,6} {2,5,7} {3,4,7} {3,5,6}
9	{1,2,3} {1,2,4} {1,2,5} {1,3,4} {1,3,5} {1,4,5} {2,3,4} {2,3,5} {2,4,5} {3,4,5}
10	{1,2,3} {1,2,4} {1,2,5} {1,3,4} {1,3,6} {1,4,5} {1,5,6} {2,3,5} {2,4,6} {3,4,5}
11	{1,2,3} {1,2,4} {1,2,5} {1,3,4} {1,3,6} {1,4,5} {2,3,4} {2,3,5} {2,4,6} {3,4,5}
12	{1,2,3} {1,2,4} {1,2,5} {1,3,5} {1,3,6} {1,4,6} {2,3,4} {2,3,6} {2,5,6} {3,4,5}
13	{1,2,4} {1,2,5} {1,2,6} {1,3,4} {1,3,5} {1,3,6} {2,3,4} {2,3,5} {2,3,6} {4,5,6}
14	{1,2,3} {1,2,4} {1,3,6} {1,4,5} {1,5,6} {2,3,5} {2,4,6} {2,5,6} {3,4,5} {3,4,6}
15	{1,2,3} {1,2,4} {1,2,5} {1,3,4} {1,3,6} {1,4,7} {2,3,4} {2,3,7} {2,4,6} {3,4,5}
Generating sets for all nonisomorphic families in $MI^2(v,4,2)$, for $v \geq 8$	
i	H^i
1..15	one application of Construction 3.3.1 to G^i
16	{1,2,7} {1,3,4,7} {1,5,6,7} {2,3,6,7} {2,4,5,7} {1,2,3,5} {1,2,4,6}
17	{1,2,3,7} {1,4,5,7} {2,4,6,7} {3,5,6,7} {1,2,5,6} {1,3,4,6} {2,3,4,5}
Generating sets for all nonisomorphic families in $MI^2(v,t+2,t)$, for $t \geq 3$, $v \geq t+6$	
i	L^i
1..17	$(t-2)$ applications of Construction 3.3.1 to H^i

Table 3.1: All generating sets for $s = 2$ and $k = t + 2$.

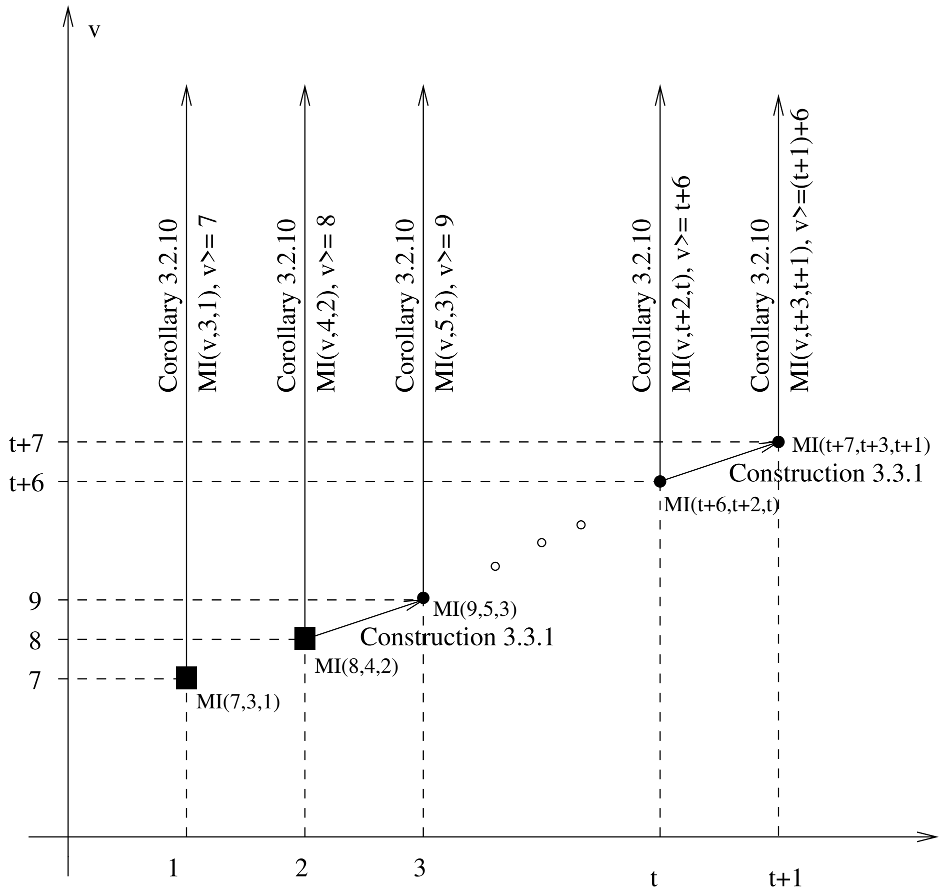

Figure 3.2: Determination of all maximal families for $s = 2$, $k = t + 2$, for all $t \geq 1$ (see Theorem 3.4.9).

essential pair for $\overline{x} \in [7, v]$, must be equivalent to one of the following cases: $C_1 = \{1, 3, 5, \overline{x}\}$, $D_1 = \{1, 2, 4, \overline{x}\}$; $C_2 = \{1, 3, 5, \overline{x}\}$, $D_2 = \{1, 4, 6, \overline{x}\}$; or $C_3 = \{1, 3, 5, \overline{x}\}$, $D_3 = \{2, 3, 6, \overline{x}\}$. For $i = 1, 2, 3$, we analyze the case of $e(\overline{x})$ joining w_{C_i} and w_{D_i}, the vertices corresponding to C_i and D_i, respectively. Let $\mathcal{G}_i$ be the subgraph of $\mathcal{G}$ that is induced by w_{C_i}, w_{D_i} and every vertex w that have noncoloured edges incident to both w_{C_i} and w_{D_i}. The edges in E' are coloured and are contained in $\mathcal{G}_i$ for some $1 \leq i \leq 3$. In Figure 3.1, we show graphs $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$ with thick edges representing the coloured edges. By inspection, we conclude that $|E'| \leq 2$. $\Box$

The following theorem is a combination of many of the previous results.

Theorem 3.4.9 *The nonisomorphic families in* $MI^2(8, 4, 2)$ *and in* $MI^2(7, 3, 1)$ *com-*

pletely determine all nonisomorphic families in $MI^2(v, t+2, t)$*, for all* $v \geq t+6$ *and* $t \geq 1$.

Proof. Suppose we have listed all nonisomorphic families in $MI^2(8,4,2)$ and $MI^2(7,3,1)$. We outline how to obtain all nonisomorphic sets in $MI^2(v, t+2, t)$ for arbitrary $v \geq t+6$, $t \geq 1$.

If $t = 1$, by our assumptions, we have listed all nonisomorphic families in $MI^2(t+6, t+2, t) = MI^2(7,3,1)$. If $t \geq 2$, perform $t-2$ applications of Construction 3.3.1 to each of the nonisomorphic sets in $MI^2(8,4,2)$. By Theorem 3.3.5 and Theorem 3.4.6, this gives all nonisomorphic sets in $MI^2(8+(t-2), 4+(t-2), 2+(t-2)) = MI^2(t+6, t+2, t)$.

By Theorem 3.4.8, we see that $n_0(t+2, t) = t+6$. Corollary 3.2.10 implies that computing generating sets $g'(\mathcal{A})$, as defined in (3.5), for all nonisomorphic sets in $MI^2(t+6, t+2, t)$ gives generating sets for all nonisomorphic sets in $MI^2(v, t+2, t)$, for $v \geq t+6$. □

Corollary 3.4.10 *Let* $t \geq 1$*, and* $v \geq t+6$*. Then,*

$$D(MI^2(v, t+2, t)) = \begin{cases} D(MI^2(7,3,1)) = 15, & \text{if } t = 1, \\ D(MI^2(8,4,2)) = 17, & \text{if } t \geq 2. \end{cases}$$

A complete list of nonisomorphic sets in $MI^2(8,4,2)$ and in $MI^2(7,3,1)$ is given in Table 3.1.

Chapter 4

Polyhedral Algorithms for Packings and Designs

Computational methods have been important in combinatorial design theory. First, they are useful for constructing "starter" designs to be used in recursive constructions of infinite families. Second, they have been employed for constructing complete lists of designs with certain parameters. Such lists can be used by researchers in order to discover properties, test hypotheses, formulate conjectures, etc. [71]. Third, exhaustive searches can be used for proving that some designs do not exist. A remarkable example of the latter is the computational proof that a projective plane of order 10, i.e. a 2–$(111, 11, 1)$ design, does not exist [61].

Techniques that have been widely used include backtracking, several local search methods (such as hill-climbing, simulated annealing, genetic algorithms) as well as several algorithms using tk-matrices ([54] [56] [58] [59] [60] [69] [90]). For a collection of articles on computational methods see the book edited by Wallis [102] and the surveys by Gibbons [43] and Mathon [71]. Recently, the first polyhedral algorithm for 2-designs was implemented using the point-block incidence matrix formulation [103]. In [76], we proposed a branch-and-cut approach for finding t-designs, packings and coverings using the tk-matrix formulation. In this chapter, we describe an implementation of this approach for finding t-designs and packings. This algorithm can be employed for the first and third applications described in the previous paragraph, namely searching for a design with given parameters or proving that such a design does not exist.

The main contribution of this chapter is the design, implementation and experimental analysis of a branch-and-cut algorithm for t–$(v, k, 1)$ designs and packings. The chapter is organized as follows. In Section 4.1, we describe the branch-and-cut algorithm, which is the most popular polyhedral method for solving integer programming problems; we also discuss briefly how some common requirements on the designs can be incorporated into this framework. In Section 4.2, we give details specific to our branch-and-cut method, including separation algorithms, initial fixings, partial isomorph rejection and other implementation issues. In Section 4.3, we exhibit and discuss our computational findings, analyzing how various parameters and variations affect the performance of the algorithm. We also include tables of cyclic packings found by our algorithm.

4.1 The polyhedral approach and design problems

4.1.1 The branch-and-cut method

Branch-and-cut is a general technique for solving integer programming problems, which has become quite popular. See the paper by Caprara and Fischetti [16] for an annotated bibliography. Remarkable success in solving large scale symmetrical traveling salesman problems is described in [5] [84].

The method combines branch-and-bound and cutting-plane algorithms. In short, a cutting-plane algorithm is employed at every node of the branch-and-bound tree. This either leads to a solution of the subproblem or to a so called *tailing-off* phenomenon (many iterations with little improvement towards integrality). In the latter case, the subproblem can be substituted by two new subproblems, by creating a new branch in the tree.

Branch-and-bound and cutting planes have been combined previously to the introduction, by Padberg and Rinaldi [83], of the term “branch-and-cut”. The distinctive feature of branch-and-cut is that cuts (i.e. valid inequalities for the problem’s polytope) are not only added to the original problem, but at every node of the branch-and-bound tree.

First, we review the cutting-plane method, then we describe the general branch-and-cut algorithm in more detail. The reader who is familiar with the approach may go directly to the next sections, in which we discuss issues on our specific application.

Suppose we have a polyhedral description of a combinatorial optimization problem by means of a large number of facet inducing inequalities, so that it would be impractical to solve the linear programming problem using all inequalities. Cutting-plane algorithms add inequalities one by one and often obtain an optimal solution before generating all of them. A cutting-plane algorithm is outlined below.

Algorithm: a general cutting-plane method

Let P_I be the polyhedron for the problem we want to solve. For any polyhedron $Q \supseteq P_I$, let $LP(Q)$ be the problem of finding $max\{c^T x : x \in Q\}$ and $x_{opt}(Q)$ be any optimal vertex solution of $LP(Q)$.

Input: a polytope P_R containing P_I.

Output:

(success, x_{opt}): $x_{opt} = x_{opt}(P_I)$, or

(failure, x_T, P_T): P_T is s.t. $P_R \supseteq P_T \supseteq P_I$, $x_T = x_{opt}(P_T)$.

$P^0 = P_R$; $k = 0$;

repeat

  compute $x^k = x_{opt}(P^k)$;

  if $x^k \in P_I$ then "optimum was found";

  else

    solve the separation problem:

      find $d \in \mathbb{R}^n, d_0 \in \mathbb{R}$ such that $d^T x \leq d_0$ is valid for P_I and $d^T x^k > d_0$;

    if "separation was successful" then

      $P^{k+1} = P^k \cap \{x \in \mathbb{R}^n : d^T x \leq d_0\}$;

      $k = k + 1$;

until "optimum was found" or "separation was not successful"

if "optimum was found"

then **return**(success, x^k);

else **return**(failure, x^k, P^k).

Remark: If an integer programming formulation is used with $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ and $P_I = conv\{x \in P : x \text{ is integer}\}$ then P_R may be set as P and the condition "$x^k \in P_I$" is equivalent to "x^k is integer". Moreover, if each inequality $d^T x \leq d_0$ induces a facet of P_I then the algorithm halts in a finite number of steps.

Cutting-plane algorithms alone may not be enough to efficiently solve a combinatorial optimization problem. It may happen that a complete description of the polyhedron is not available, that the separation problem cannot be efficiently solved for a class of inequalities or that the above mentioned *tailing-off* phenomena arises. So, the cutting-plane algorithm is combined with the branch-and-bound method.

Branch-and-bound is an implicit enumeration method. Let us focus on 0-1 integer programming problems. A binary tree is associated with the solution of the problem. The root of the tree represents the original relaxed problem (i.e. the problem obtained from the integer programming problem by dropping the integrality requirements). The left child and right child of a node N correspond to subproblems of the problem associated with N obtained by fixing a free variable at 0 or 1, respectively. A list $\mathcal{L}$ of current nodes to be processed is kept. The list $\mathcal{L}$ is initialized with the root node. At each step of the algorithm, a node is removed from $\mathcal{L}$ and processed. The processing of a node N starts by solving the relaxed subproblem corresponding to N. The node N is pruned, i.e., the subtree rooted at N is neglected, if one of the following cases occur: the subproblem is infeasible, the solution for the subproblem is integral, or the solution of the subproblem is not integral but the optimal value is less than or equal to the objective value of the best solution found. In all other cases, the children of node N are added to $\mathcal{L}$. The algorithm terminates when $\mathcal{L}$ is empty, and returns the best integral solution found as the optimal solution.

The branch-and-cut algorithm differs from pure branch-and-bound in that at each node it applies a cutting-plane algorithm rather than simply solving the relaxed sub-problem. In addition, the cuts generated in one node of the tree can be reused in another node by keeping a pool of inequalities.

The design of a branch-and-cut algorithm involves many algorithmic and implementation decisions, several of which depend on the problem in consideration. We describe our implementation in Section 4.2.

4.1.2 Branch-and-cut and combinatorial design problems

Besides being an alternative to tackle combinatorial design problems, the branch-and-cut approach offers other advantages. We outline how to adapt the branch-and-cut algorithm to deal with specific design questions:

1. **Extending designs.** By extending a design, we mean the well-known process in which a point is added to every block of a t-(v, k, λ) design and extra blocks are added in order to form a $(t+1)$–$(v+1, k+1, \lambda)$ design. In many situations, design theorists are concerned whether a design can be extended to a larger design in this fashion. Little modification in a branch-and-cut algorithm is required for producing extensions: just fix to 1 the variables corresponding to the blocks arising from the smaller design.

2. **Fixing subdesigns.** This is a generalization of the previous feature. Here, any number of blocks can be selected to be in the design.

3. **Forbidding subconfigurations.** As exemplified by the anti-Pasch designs, forbidding subconfigurations can be accomplished by adding extra inequalities. Although the number of such inequalities might be too large to be incorporated into the original problem, it is possible to generate the violated ones as part of the cutting-plane algorithm.

4. **Assuming the action of an automorphism group.** This is accomplished, as in other tk-matrix methods, by using a different original matrix (as seen in Section 2.1.1).

5. **Proving nonexistence results.** Branch-and-cut is an implicit enumeration technique, and it is able to prove nonexistence of t-designs, whenever the upper bound

on all the active subproblems drops below $b - \varepsilon$, where b is the number of blocks in the design and ε is large enough to account for approximation errors.

4.2 A branch-and-cut implementation for t-$(v, k, 1)$ designs and packings

Our implementation handles the following types of designs and employs the following classes of inequalities:

t–$(v, k, 1)$ designs and packings	(separation of) clique inequalities	m-sparse inequalities
general	general	N/A
general with $k = t + 1$	general and specialized	N/A
4-sparse ($t = 2$, $k = 3$)	general and specialized	$m = 4$
5-sparse ($t = 2$, $k = 3$)	general and specialized	$m = 5$
with cyclic automorphism	general	N/A

Table 4.1: Types of designs and inequalities used in our implementations.

The general framework of a branch-and-cut algorithm is given in Section 4.1. In this section, we describe subalgorithms and other issues specific to our implementation.

4.2.1 Initialization

Some variable fixing can be done in the original problem of finding t-designs, before running the branch-and-cut algorithm, as we describe now.

For a t–$(v, k, 1)$ design all the k-subsets of $[1, v]$ passing through a fixed $(t-1)$-set can have their variables fixed either to zero or one. For example, for a 2–$(7, 3, 1)$ design, since $\{1\}$ has to appear exactly once together with every other element in some block, we can assume the following blocks are in the design

$$B_1 = \{1, 2, 3\} \quad B_2 = \{1, 4, 5\} \quad B_3 = \{1, 6, 7\}$$

and all the other 3-subset of $[1,7]$ containing $\{1\}$ are not present in the design. In addition, a block not containing $\{1\}$ must contain at most one element of each of B_1, B_2, B_3, so we can assume w.l.o.g. that $E = \{2,4,6\}$ is a block in the design.

In general, for any t–$(v,k,1)$ design and any given $(t-1)$-subset S of $[1,v]$, the blocks of the design that contain S are unique up to permutations. Therefore, similarly to the previous example, we can assume w.l.o.g. that

1. the following blocks are present in the design:

$$\begin{aligned} B_i &= [1,t-1] \cup [t+(i-1)(k-t+1), t+i(k-t+1)-1],\ 1 \leq i \leq \frac{v-t+1}{k-t+1}, \\ E &= [1,t-2] \cup \bigcup_{1 \leq i \leq k-(t-2)} \{t+(i-1)(k-t+1)\}; \end{aligned}$$

2. all the other subsets of $[1,v]$ containing $[1,t]$ are not present in the design.

4.2.2 Separation algorithms

At each iteration of the cutting-plane, a separation problem must be solved for each of the classes of facets being considered. An inequality $d^T x \leq d_0$ is considered to be violated whenever $d^T x - d_0$ is larger than a parameter called `VIOLATION-TOLERANCE`. This parameter has a great impact on the algorithm performance as we will discuss in Section 4.3.

Next, we discuss the separation algorithms and related issues used in our implementation.

The fractional intersecting graph

The separation of clique facets relies on finding violated cliques in the intersection graph of the original matrix. For our $W^v_{t,k}$ matrix, this graph has $\binom{v}{k}$ nodes and it would be extremely time consuming to search for cliques or other subgraphs in such a large graph. However, it is well known [50] that in any set packing problem, we can restrict our attention to the *fractional intersection graph*, i.e., the subgraph of the intersection graph induced by the fractional variables. More precisely, in any set packing problem, the support of a violated inequality cannot contain variables that are equal to 1, and the variables that are equal to 0 do not contribute to the violation; therefore, the violated inequalities can be found just by analyzing the fractional variables and lifting the inequality including possible null variables. In our experiments, the number of fractional variables is roughly three times the expected number b of blocks, i.e., around $3\left\lfloor\frac{\binom{v}{t}}{\binom{k}{t}}\right\rfloor$, in contrast with the total number of variables being $\binom{v}{k}$. This reduces the size of the graph we have to deal with from several thousand to a few hundred nodes.

In our problem each variable corresponds to a k-subset of $[1, v]$. A canonical ordering on the subsets is used, so that each subset corresponds to an integer number that is the index of the variable. The fractional intersection graph of $W^v_{t,k}$ is constructed in the following way: each fractional variable corresponds to a node, and two nodes are adjacent if their corresponding sets intersect in at least t points.

Hoffman and Padberg [50] report that decomposing the fractional graph into connected components resulted in large savings for their set partitioning problems, which arise in an airline crew scheduling application. We did implement the decomposition of the fractional graph into connected components. However, it turned out that our design theoretical problems have fractional graphs, in most of the cases, formed by a single connected component.

Separation of clique facets

Clique facets are used for all types of designs in Table 4.1. Given $\overline{x}$, we must find violated cliques in the fractional graph $G^v_{k,t}$ (corresponding to $\overline{x}$). A clique C is considered violated if $\sum_{i\in C} \overline{x}_i - 1 >$ `VIOLATION-TOLERANCE`.

`GENERAL-CLIQUE-SEPARATION`:

The general clique detection employed by our algorithm works for a general graph. We borrowed several ideas from Hoffman and Padberg [50] and Nemhauser and Sigismondi [78]. For every node v, we search for a violated clique containing v. Let $N(v)$ denote the neighborhood of v. Since every clique containing v must be contained in $v \cup N(v)$ we concentrate in this subgraph. We look at the neighborhood of a node v. If the neighborhood is small, say, under 20 nodes, we enumerate every clique in the neighborhood, select the violated ones, and lift them to the larger problem (including null variables). If the neighborhood is larger we use two greedy heuristics proposed by Nemhauser and Sigismondi [78]. Both heuristics select, among the candidate nodes to join a partial clique, the one satisfying a given criteria. The first heuristic chooses the fractional nodes with larger x_i, and the second one, the fractional node closest to 0.5. However, we observed that in most iterations the fractional variables are all under 0.5, so these two heuristics turned out to be the same for our problems.

`SPECIAL-CLIQUE-SEPARATION` (for k=t+1):
We implemented the special clique separation algorithm for designs with $k = t + 1$, given on page 42. Recall that this algorithm uses the knowledge of the clique structure for these problems and examines each edge of the graph exactly once, since there is at most one violated clique passing through each edge. Note that we use the fractional intersection graph in place of $G^v_{k,t}$, as mentioned previously.

Separation of 4-sparse and 5-sparse inequalities

Recall that m-sparse inequalities only apply to 2-$(v, 3, 1)$ designs and packings. Although the m-sparse inequalities are part of the original problem formulation, we do not add them to the original problem due to the large number of inequalities of this type; there are $\binom{v}{6}$ 4-sparse inequalities and $\binom{v}{6} + \binom{v}{7}$ 5-sparse inequalities. Instead, we only add m-sparse inequalities that are violated during the cutting-plane algorithm.

We implemented separation algorithms for 4-sparse and 5-sparse inequalities. Recall that the 4-sparse inequalities are of the form in (4.1) and 5-sparse inequalities are of the form in (4.1) or (4.2):

$$\sum_{K \in \binom{T}{3}} x_K \leq 3, \qquad \text{for } T \subseteq [1, v], |T| = 6; \tag{4.1}$$

$$\sum_{K \in \binom{L}{3}} x_K \leq 4, \qquad \text{for } L \subseteq [1, v], |L| = 7. \tag{4.2}$$

A separation algorithm could go over all 6-subsets or 7-subsets of $[1, v]$ and check whether the corresponding inequalities (4.1) and (4.2) are violated. However, this would be very inefficient due to the large number of possible inequalities most of which would not be violated.

Our separation algorithm employs the following idea. First, we determine every set with 3 triples (corresponding to nonzero variables) whose union spans a 6-subset and such that every pair of triples intersects in exactly one point. Then, we check whether the corresponding inequalities of type (4.1) are violated. Similarly, we determine all sets of 3 triples (corresponding to nonzero variables) spanning a 7-set and such that every pair of triples intersects in exactly one point, and check whether the corresponding inequalities of type (4.2) are violated. These conditions are not guaranteed to find all the violated inequalities for fractional points. For example, consider a fractional point with $x_K = 0.33$ for all triples $K \subseteq [1,5]$, and $x_K = 0$ for all other triples in $[1,6]$; then an inequality of type (4.1) is violated for $T = [1,6]$, but there are no 3 triples (with positive values) contained in $[1,6]$ that span $[1,6]$. However, when the variables are integer, inequality (4.1) is violated if and only if a Pasch is present, in which case we know any 3 sets in the Pasch span the 6-set and are pairwise intersecting. Moreover, as we progress towards integrality, it is more likely that this condition will suffice to detect violated 4-sparse inequalities. Similarly, 5-sparse inequalities are not guaranteed to be found by this method, but when integrality is attained, these conditions guarantee the detection of violated 5-sparse inequalities.

Let us now describe the method used for detecting the above mentioned conditions, i.e. for detecting the groups of 3 triples with every pair intersecting in a point and spanning either a 6-subset or a 7-subset of $[1,v]$.

First, we construct a 1-intersecting graph of nonzero variables, i.e. a graph whose vertices correspond to nonzero variables and such that two nodes are connected if the corresponding triples intersect in exactly one point. Then, we go over each 3-cycle in the graph and check whether the union of the corresponding triples spans 6 or 7 points. In order to avoid finding each 3-cycle more than once, the edges of the graph are directed from the smaller triple to the larger one (smaller here, with respect to the canonical ordering of triples). The algorithm searches for 3 edges of the type (K_1,K_2), (K_2,K_3) (K_1,K_3). The algorithm is outlined below.

Algorithm: separation of 4-sparse and 5-sparse inequalities.

Construct the direct 1-intersecting graph of nonzero variables:

Let $\overline{x} \in \mathbb{R}^{\binom{v}{3}}$ be the point to be separated.

create a node for each K with $\overline{x}_K > 0$.

for every pair of nodes K_1 and K_2 do

if $|K_1 \cap K_2| = 1$ then

create a directed edge from $\min(K_1, K_2)$ to $\max(K_1, K_2)$.

for every directed edge (K_1, K_2)

for every node K_3 such that (K_2, K_3) and (K_1, K_3) are edges do

take $S = K_1 \cup K_2 \cup K_3$.

if $|S| = 6$ then

if $\sum_{K \subset \binom{S}{3}} \overline{x}_K - 3 >$ `VIOLATION-TOLERANCE`, then store $(S, 3)$

else

if $|S| = 7$ then

if $\sum_{K \subset \binom{S}{3}} \overline{x}_K - 4 >$ `VIOLATION-TOLERANCE`, then store $(S, 4)$

Pool of cuts

Previously generated inequalities are stored in a pool of cuts in order to reuse them in other nodes of the tree. This is a common feature in a branch-and-cut algorithm (see [16]).

Any cut that is generated is stored in the pool of cuts. Before running the separation algorithm for various facets we check whether some inequalities from the pool are violated. If the number of violated inequalities found in the pool is larger than a certain value, there is no need of calling the separation routines.

The data structure for the pool of cuts is a hash table and a linked list of inequalities. The entries of the hash table are pointers to inequalities and collisions are resolved by chaining. The hash function is given by the sum of the square of the variable indices corresponding to nonzero coefficients in the inequalities modulo the table size. Before inserting a new constraint on the list, we check if it is already present through the pointer in the hash table. To check if there are violated inequalities in the pool, the algorithm searches linearly along the linked list.

4.2.3 Criteria for abandoning the cutting-plane algorithm

Our algorithm stops cutting-plane iterations if any one of the following conditions is satisfied: the optimal solution to the LP relaxation is integral (i.e. the subproblem rooted at the node was solved), the subproblem is infeasible, or the addition of the cuts is "not producing much improvement". The last condition is measured by the number of cuts and the "quality" of cuts at the previous iteration. The cutting-plane algorithm is abandoned whenever the number of cuts in the previous iteration is smaller than a parameter `MIN-NUMBER-OF-CUTS` or the maximum violation is smaller than a parameter `MIN-WORTHWHILE-VIOLATION`.

4.2.4 Partial isomorph rejection

For combinatorial design problems, several subproblems in the branch-and-cut tree may be equivalent. Recall that a node in the branch-and-cut tree corresponds to the subproblem in which the variables in the path from the root to the node have their values fixed either to zero or one. Let N be any node of the tree and denote by $\mathcal{F}_0(N)$ and $\mathcal{F}_1(N)$ the collection of blocks corresponding to the variables fixed to 0 and 1, respectively, in the path from the root of the tree to N. If N and M are nodes in the tree with $(\mathcal{F}_0(N), \mathcal{F}_1(N))$ isomorphic to $(\mathcal{F}_0(M), \mathcal{F}_1(M))$ then equivalent problems are going to be unnecessarily solved. The partial isomorph rejection we describe in this section aims at reducing the number of such equivalent subproblems.

Let N be a node of the branch-and-cut tree with two children N_0 and N_1. The original branching scheme would make N_0 correspond to $x_K = 0$ and N_1 correspond to $x_K = 1$, for some variable K. This branching scheme is modified so that the number of nodes in the tree is reduced by avoiding some subproblems that are equivalent to others already considered, as we discuss next.

Let A be the permutation group acting on $[1, v]$ that fixes $\mathcal{F}_0(N)$ and $\mathcal{F}_1(N)$, and let $A(K)$ be the orbit of K under A. The new branching scheme for partial isomorph rejection lets N_0 correspond to "$x_L = 0$ for all $L \in A(K)$" and N_1 correspond to "$x_L = 1$ for some $L \in A(K)$". The tree reduction comes from letting N_1 correspond, w.l.o.g., to "$x_K = 1$" instead. Thus the new branching scheme implies

$$\begin{aligned} \mathcal{F}_0(N_0) &= \mathcal{F}_0(N) \cup A(K), & \mathcal{F}_1(N_0) &= \mathcal{F}_1(N), \\ \mathcal{F}_0(N_1) &= \mathcal{F}_0(N), & \mathcal{F}_1(N_1) &= \mathcal{F}_1(N) \cup \{K\}. \end{aligned}$$

This new branch-and-cut tree has at most as many problems as the regular one, since when $|A(K)| > 1$ not only K but other variables in $A(K)$ are being simultaneously fixed at 0 in N_0.

All we need is an algorithm that, given collections $\mathcal{F}_0$ and $\mathcal{F}_1$ of k-subsets of $[1, v]$ and a k-subset K of $[1, v]$, computes

1. the permutation group A acting on $[1, v]$ that fixes $\mathcal{F}_0$ and $\mathcal{F}_1$;

2. the orbit $A(K)$ of K under A.

The first problem is equivalent to finding the permutation group acting on the vertices of a special graph that fixes some subsets of the vertices. Consider the bipartite graph $G_{[1,v],\mathcal{F}_0 \cup \mathcal{F}_1}$ whose vertex partition corresponds to points in $[1, v]$ and sets in $\mathcal{F}_0 \cup \mathcal{F}_1$, and such that $p \in [1, v]$ is connected to $F \in \mathcal{F}_0 \cup \mathcal{F}_1$ if and only if $p \in F$. Thus, our problem is equivalent to finding the permutation group acting on the vertices of the graph that fixes vertices in $[1, v]$, in $\mathcal{F}_0$ and in $\mathcal{F}_1$. This can be computed using the package *Nauty*, by Brendan McKay, the "most powerful general purpose graph isomorphism program currently available" (quoted from [52]). For a description of the algorithms used by *Nauty* see [73].

The second problem can be solved by a simple algorithm that we describe now. A collection $\mathcal{S}$ of k-sets is initialized with K. At every step, a different set L in $\mathcal{S}$ is considered and, for all $\pi \in A$, the set $\pi(L)$ is added to $\mathcal{S}$. The algorithm halts when all sets in $\mathcal{S}$ have been considered, and thus, $A(K) = \mathcal{S}$.

A small variation of the two previous methods can dramatically improve efficiency when $|\mathcal{F}_0 \cup \mathcal{F}_1| << v$. Let $R = \cup_{\mathcal{B} \in \mathcal{F}_0 \cup \mathcal{F}_1} \mathcal{B}$. Consider the graph $G_{R,\mathcal{F}_0 \cup \mathcal{F}_1}$ instead of $G_{[1,v],\mathcal{F}_0 \cup \mathcal{F}_1}$ and apply the method described above to compute an automorphism group

A'. The points in $[1,v] \setminus R$ are isolated vertices in $G_{[1,v],\mathcal{F}_0 \cup \mathcal{F}_1}$, and therefore form a cycle in any permutation in A. Thus A' is the restriction of A to points in R. In order to compute $A(K)$, we use the method described above and compute $A'(K \cap R)$, and then compute $A(K)$ by taking all the k-subsets of $[1,v]$ that contain some set in $A'(K \cap R)$. Two kinds of improvements in efficiency are observed. In the first part, the original graph gets reduced by $[1,v] \setminus R$ nodes. In the second part, if $|K \cap R| < k$ the size of the set $\mathcal{S}$ in the second problem is reduced by a factor of $\binom{[1,v]\setminus R}{k-|K\cap R|}$.

4.2.5 Branch-and-cut tree processing

The main strategies to be defined in the branch-and-cut node processing are: the selection of the variable involved in the branching at a node and the selection of the next node to be processed.

We implemented two strategies for the selection of the branching variable in a node. The first one selects the variable with largest fractional value, and the second one, the variable closest to 0.5. Both strategies turned out to be equivalent, since in our problems most of the fractional variables are smaller than 0.5.

Typically, the objective value of the relaxed problem and of the integer problem are the same or quite close in our design problems. This suggests that a tree search guided by objective value is often meaningless. Our selection of the next node to process is done as a depth-first search, giving priority to nodes with variables fixed to 1. The rationale behind this heuristic can be explained as follows. In general, fixing a variable to 1 makes a packing problem tighter than fixing it to 0. Depth-first search is employed in order to fix as many variables to 1 as possible in an attempt to quickly find a design.

4.2.6 Other implementation issues

Preprocessing is a technique that allows for a reduction on the size of a linear programming problem. It involves several rules for variable and constraint elimination through the testing of simple conditions. Preprocessing has proved to have a great impact on efficiently solving integer programming problems [30]. No preprocessing rule, besides the initial fixing mentioned in Section 4.2.1, was implemented. However, most of the conditions that could allow for problem reduction do not hold initially in our problems. For example, our problems have no empty rows or columns, the problems are given by a matrix of cliques, they contain no dominated rows or columns, etc. On the other hand, such conditions might hold deeper in the tree, and preprocessing could become advantageous there. This is a direction for further research. For general discussions on preprocessing techniques see the article by Crowder, Johnson and Padberg [30] or the book by Nemhauser and Wolsey [80, pages 17–20], and for such a study in the context of set partitioning problems see the thesis by Borndörfer [13] and the references therein.

Objective function perturbation is a technique based on adding small (randomly chosen) quantities to each of the objective function coefficients. This is done in such a way that the optimal solution for the perturbed problem remains the same as the original one. This technique sometimes works well for problems with many optimal solutions, by making some of them more attractive, leading the linear programming algorithm directly towards them. We experimented with this technique and, although in some cases we got a reduction on the number of nodes in the tree, the linear programming solution at each node was more expensive, which led to a greater overall time.

4.3 Computational results

In this section, we report on computational experiments with the branch-and-cut implementation described in the previous section. The following aspects are investigated.

1. The kinds and sizes of problems our implementation can handle.

2. The effect of isomorphism rejection for t-designs and packings, including the cases in which the designs do not exist.

3. The impact of cutting: a comparison between branch-and-cut and branch-and-bound.

4. A comparison between the specialized and the general clique separation algorithms for t-designs and packings with $k = t + 1$.

5. The influence of parameters that affect the trade-off between branching and cutting, namely `VIOLATION-TOLERANCE` and `MIN-WORTHWHILE-VIOLATION`.

6. Changes on the algorithm's performance given by additional properties such as m-sparse conditions.

7. A comparison of our branch-and-cut with a leading general purpose branch-and-bound software.

The tests are run on a Sun Ultra 2 Model 2170 workstation with 245 MB main memory and 1.2 GB virtual memory, operating system SunOS 5.5.1. Our branch-and-cut consists of about 10,000 lines of code written in C++ language and compiled with g++ compiler. The following packages are linked with our code: LEDA Library version 3.2.3 [93] for basic data structures such as lists and graphs, CPLEX package version 4.0.8 [29] for solving linear programming subproblems, and Nauty package[1] version 2.0 [73] for finding automorphism groups of graphs.

In the following we discuss our experimental results. We ran the algorithm for several designs and packing parameters which are described in Tables 4.2 and 4.3.

We explain the columns in the tables describing the experiments. The first few columns indicate problem parameters, and they vary from table to table. The other columns are described as follows. Columns (BB) and (BT) give the number of explored branch-and-bound tree nodes and the number of times the algorithm backtracks, respectively. Columns (IE), (IR), (Mxl) and (IT) encode information on the isomorph rejection algorithm. Column (IE) gives the number of times the algorithm is executed; (IR) gives the number of times the algorithm returned a nontrivial orbit (one with at least two sets); (Mxl) is the largest depth of a node in which a nontrivial orbit was found; (IT) is the total time spent in isomorph rejection. The next two columns (LPT) and (LPs) report on total time solving linear programming problems and the number of such problems, respectively. Column (ST) denotes total separation time; (Cl) and (Sp) indicate the number of times clique and sparse inequalities are added to a subproblem. Finally,

[1] with an adaptation by Luis Dissett for having the automorphism group as an output parameter rather than a printing.

(TotT) contains the total time. All times are measured in seconds.

In all tables, v and b denote the number of points and blocks in the design, respectively. Remember that the optimal objective value is equal to b. In the tables for packings, column (Sch) shows the Schönheim upper bound for the corresponding packing. This bound is used in the algorithm as an initial upper bound for the objective value. When applicable, column (exist.) indicates whether a certain design exists. Tables 4.8 and 4.9 have a column (run#) which correspond to the parameter combination given in Table 4.7.

In Tables 4.4 and 4.5 we report on isomorph rejection statistics. Table 4.4 compares the same instances of packings with and without isomorph rejection (specified by column (IRej)). Only the packings with $v = 5, 11, 12, 14$ requires a call to the algorithm. From these parameters, only $v = 5, 11$ benefit from nontrivial orbits, and only $v = 11$ profits from the isomorph rejection. We observe that the time spent in the isomorph rejection algorithm is very small compared to the total time. The packing for $v = 11$ could not be found without the isomorph rejection. The difficulty encountered for $v = 11$ is that the Schönheim bound is not met by the packing size. Therefore, the program might find a solution of size 17, but has to go over most of the branches to conclude it is optimal. The isomorph rejection reduces the amount of branches to be searched. We conclude that the isomorph rejection algorithm is effective since it spends little extra time and adds the benefit of tackling the hardest problems.

In Tables 4.5 and 4.6 we report on the effectiveness of using cuts, as opposed to simple branch-and-bound, for 2–$(v, 3, 1)$ designs and packings. For designs there is no clear winner in terms of total time (Table 4.5). For packings, the total time using cuts is either comparable or substantially smaller than without cuts, especially for the larger instances (Table 4.6). In all cases, using cuts reduces the number of explored tree nodes and the number of times the algorithm backtracks. This is reflected in the often smaller number of solved linear programming problems and time spend on solving them. However, the time spent on cut separation makes the cutting version worst for some instances. Larger problems should benefit from the cutting version, since linear programming tends to dominate the running time.

Tables 4.8 and 4.9 summarize the results of 8 runs corresponding to a combination of parameters described in Table 4.7. Even and odd numbered runs correspond to specialized and general separation algorithms, respectively. From these tables we observe that the specialized separation is done much faster than the general one (see column (ST)). In all runs the specialized separation produced savings in the total running time of up to 50%. These tables also show the influence of the parameters MIN-WORTHWHILE-VIOLATION and VIOLATION-TOLERANCE. Let us denote them by MWV and V, respectively. An interval $[v_{min}, v_{max}]$ is assigned to each of these parameters. The algorithm initially sets the value of the parameter to v_{max}; as the number of fractional variables decreases, the parameter is continuously reduced towards v_{min}. The 4 combinations shown in Table 4.7 are tried. The best runs involve (MWV, V) = $[(0.3, 0.3), (0.3, 0.3)]$ and $[(0.3, 0.6), (0.3, 0.6)]$. The main conclusion is that the performance is positively affected by requiring stronger cuts. In all other tables, these parameters are set as MWV = V = $[(0.3, 0.6)]$.

In Tables 4.8 and 4.9 our runs are also compared to the general purpose integer programming solver offered by CPLEX. As problem sizes grow, our algorithm runs much faster than CPLEX (2 to 10 times faster in the two largest problems). This comparison is only included as a reference. Other problems such as m-sparse cannot be solved by such a general package since the number of inequalities defining the integer programming problem is very large. We only manage to solve some instances of these problems, by adding only the violated m-sparse inequalities in the cutting-plane algorithm.

Tables 4.10 and 4.11 show the solution of 4-sparse and 5-sparse Steiner triple systems. For 4-sparse systems we observe that $v = 13$ is harder than $v = 15$. In fact, for $v = 13$ the program has to establish its nonexistence, which is time consuming. However, the same instance $v = 13$, for which a 5-sparse system is also nonexistent, can be solved more efficiently due to the addition of 5-sparse facets. This makes the algorithm detect its nonexistence earlier. The algorithm encounters difficulties in solving 4-sparse and 5-sparse problems much earlier than when solving ordinary Steiner triple systems. For 4-sparse and 5-sparse problems the number of tree nodes grows immensely starting on $v = 19$ and $v = 15$, respectively. The second part of both tables indicates problems that did not finish after 2 hours of running time. Our conclusion is that the m-sparse restriction imposes extra difficulties for this method. Further research should be done in order to solve larger $4,5$-sparse problems.

Steiner quadruple systems are shown in Table 4.12. The case $v = 14$ is already a hard instance. In earlier versions of this implementation it took about 6 hours to solve this problem. Currently, it takes about 40 minutes to solve it. The next instance $v = 16$ is still a challenge for this implementation. Although the design is known to exist, other computational methods also fail to find this design.

Tables 4.13-4.21 report on cyclic designs and packings. Recall that these problems are solved by assuming a cyclic automorphism group action on the design, which produces reductions of the problem size, as shown in Tables 4.2 and 4.3. Tables 4.14-4.17 compare the size of maximal t–$(v, k, 1)$ packings to the size of regular packings for $t = 2, 3, 4, 5$, $k = t + 1, t + 2$, and small v. Columns (Bl) and (C) indicate the number of base blocks and the total number of blocks in the cyclic packings, respectively. To the best of our knowledge, this is the first time this quantities are computed. In columns (D) and (S) we include known values for the size of a maximal ordinary packing and Schönheim upper bounds, respectively. Thus, we must have C $\leq$ D $\leq$ S. Observe that in most cases C is not much smaller than D (or not much smaller than S in the cases that D is unknown, see Tables 4.16 and 4.17). In Tables 4.18 and 4.20, we show a sample of statistics on some runs for cyclic packings; their corresponding base blocks are shown in Tables 4.19 and 4.21. The packings can be obtained by developing the blocks modulo v. Our experiments show that the sizes of cyclic packings are close to maximal ordinary packings (compare (C) and (D)), and they are much easier to compute and compact to store. Therefore, these objects should be very attractive for applications. The size of a maximal cyclic packing also offers a good lower bound for the packing number. Cyclic Steiner triple systems are related objects, which have been given some attention (see for instance [85]).

Our previous analysis outlines the most efficient strategies and variations. The results show that the specialized clique separation greatly improves the running time. The partial isomorph rejection algorithm is also effective given the little time it spends and the great deal of savings it offers for hard problems. In addition, the influence of other parameters is discussed. The branch-and-cut algorithm can be applied to a wide range of designs, as our experiments have shown. Cyclic packings seem to be interesting objects which deserve further investigation.

		ordinary designs				cyclic designs	
t–$(v,k,1)$		before fixing		after fixing			
design	b	#rows	#cols	#rows	#cols	#rows	#cols
2-(7,3,1)	7	21	35	15	21	3	2
2-(9,3,1)	12	36	84	28	55	4	7
2-(13,3,1)	26	78	286	66	219	6	16
2-(15,3,1)	35	105	445	91	363	7	25
2-(19,3,1)	57	171	969	153	815	9	42
2-(21,3,1)	70	210	1330	190	1139	10	55
2-(25,3,1)	100	300	2300	286	2023	12	80
2-(27,3,1)	117	351	2925	325	2599	13	97
2-(31,3,1)	155	495	4495	465	4059	15	130
2-(33,3,1)	176	528	5456	496	4959	16	151
3-(8,4,1)	14	56	70	50	54	7	8
3-(10,4,1)	30	120	210	112	181	12	19
3-(14,4,1)	91	364	1001	352	934	26	67
3-(16,4,1)	140	560	1820	546	1728	35	110

Table 4.2: Problem sizes and statistics for t-designs

t-(v,k,1)	ordinary packings		cyclic packings	
packings	#rows	#cols	#rows	#cols
2-(5,3,1)	10	10	2	0
2-(6,3,1)	15	20	3	1
2-(7,3,1)	21	35	3	2
2-(8,3,1)	28	56	4	2
2-(9,3,1)	36	84	4	7
2-(10,3,1)	45	120	5	4
2-(11,3,1)	55	165	5	10
2-(12,3,1)	66	220	6	11
2-(13,3,1)	78	286	6	16
2-(14,3,1)	91	364	7	14
3-(6,4,1)	20	15	4	1
3-(7,4,1)	35	35	5	2
3-(8,4,1)	58	70	7	8
3-(9,4,1)	84	126	10	9
3-(10,4,1)	120	210	12	19
3-(11,4,1)	165	330	15	25
3-(12,4,1)	220	495	19	37
3-(13,4,1)	286	715	22	49
3-(14,4,1)	364	1001	26	67
3-(15,4,1)	455	1365	31	82
3-(16,4,1)	560	1820	35	110
3-(17,4,1)	680	2380	40	132
3-(18,4,1)	816	3060	46	160
3-(19,4,1)	969	3876	51	195
3-(20,4,1)	1140	4845	57	238
3-(21,4,1)	1330	5985	64	270
3-(22,4,1)	1540	7315	70	325
4-(8,5,1)	70	56	10	2
4-(9,5,1)	126	126	14	9
4-(10,5,1)	210	252	22	18
4-(11,5,1)	330	462	30	37
4-(12,5,1)	495	792	43	52
4-(13,5,1)	715	1287	55	93
4-(14,5,1)	1001	2002	73	122

Table 4.3: Problem sizes and statistics for packings

v	b	Sch	IRej	BB	BT	IE	IR	Mxl	IT	LPT	LPs	ST	Cl	TotT
5	2	3	yes	3	1	1	1	0	0	0	3	0	0	0.01
5	2	3	no	7	3	0	0	-	0	0.02	8	0	1	0.02
6	4	4	yes	3	0	0	0	-	0	0	3	0.01	0	0.01
6	4	4	no	3	0	0	0	-	0	0.01	3	0	0	0.01
7	7	7	yes	1	0	0	0	-	0	0.01	1	0	0	0.02
7	7	7	no	1	0	0	0	-	0	0.01	1	0	0	0.02
8	8	8	yes	5	0	0	0	-	0	0.02	8	0.02	6	0.06
8	8	8	no	5	0	0	0	-	0	0.02	8	0.02	6	0.05
9	12	12	yes	2	0	0	0	-	0	0.02	3	0	1	0.04
9	12	12	no	2	0	0	0	-	0	0.02	3	0.02	1	0.05
10	13	13	yes	11	0	0	0	-	0	0.08	14	0.05	4	0.14
10	13	13	no	11	0	0	0	-	0	0.05	14	0.08	4	0.13
11	17	18	yes	649	324	324	11	10	0.26	6.41	829	2.46	319	9.94
11	17	18	no	-	-	-	-	-	-	-	-	-	-	(*)
12	20	20	yes	17	1	1	0	-	0	0.22	24	0.25	11	0.53
12	20	20	no	17	1	0	0	-	0	0.24	24	0.27	11	0.53
13	26	26	yes	13	0	0	0	-	0	0.12	14	0.34	1	0.48
13	26	26	no	13	0	0	0	-	0	0.13	14	0.33	1	0.49
14	27	28	yes	77	27	27	0	-	0.05	1.82	121	0.85	86	2.91
14	27	28	no	77	27	0	0	-	0	1.81	121	0.88	86	2.93

(*) the algorithm failed to find the designs even after exploring 400,000 branches.

Table 4.4: The effect of partial isomorph rejection on 2-$(v, 3, 1)$ packings

b:	number of blocks	BB:	number of explored B&B nodes
Sch:	Schönheim bound	BT:	number of backtracks
IRej:	isomorph rejection algorithm	LPT:	total time solving LP problems
IE:	number of calls to IRej	LPs:	number of LP problems
IR:	number of successfull IRej's	ST:	total time in separation alg.
Mxl:	Max. depth of node in successfull IRej	Cl:	total number of added cliques
IT:	Total time in IRej	TotT:	total time

v	b	Cuts	BB	BT	IE	IR	Mxl	IT	LPT	LPs	ST	Cl	TotT
7	7	yes	1	0	0	0	-	0	0	1	0	0	0
7	7	no	1	0	0	0	-	0	0	1	0	0	0.01
9	12	yes	1	0	0	0	-	0	0.01	1	0	0	0.01
9	12	no	1	0	0	0	-	0	0.01	1	0	0	0.01
13	26	yes	6	0	0	0	-	0	0.07	8	0.1	4	0.18
13	26	no	40	17	17	0	-	0.14	0.38	40	0	0	0.57
15	35	yes	1	0	0	0	-	0	0.06	1	0	0	0.08
15	35	no	1	0	0	0	-	0	0.06	1	0	0	0.08
19	57	yes	35	1	1	0	-	0.02	1.45	39	2.66	5	4.17
19	57	no	112	42	42	0	-	0.78	5.2	112	0	0	6.68
21	70	yes	58	8	8	0	-	0.23	3.86	63	6.17	7	10.29
21	70	no	57	12	12	0	-	0.34	3.02	57	0	0	3.81
25	100	yes	60	0	0	0	-	0	9.96	69	17.27	12	27.44
25	100	no	74	6	6	0	-	0.29	16.19	74	0	0	16.99
27	117	yes	76	4	4	0	-	0.26	32.61	89	29.89	16	63.09
27	117	no	290	111	111	0	-	6.28	70.83	290	0	0	85.71
31	155	yes	113	7	7	0	-	0.73	87.61	127	76.35	25	165.56
31	155	no	358	128	128	0	-	13.95	277.69	358	0	0	311.52
33	176	yes	174	26	26	0	-	3.31	309.52	198	114.61	53	430.36
33	176	no	267	71	71	0	-	10.46	209.06	267	0	0	235.84

Table 4.5: Branch-and-cut versus branch-and-bound for 2-$(v, 3, 1)$ designs

v	b	Cuts	BB	BT	IE	IR	Mxl	IT	LPT	LPs	ST	Cl	TotT
5	2	yes	3	1	1	1	0	0	0	3	0	0	0.01
5	2	no	3	1	1	1	0	0	0	3	0	0	0.01
6	4	yes	3	0	0	0	-	0	0	3	0.01	0	0.01
6	4	no	3	0	0	0	-	0	0.01	3	0	0	0.01
7	7	yes	1	0	0	0	-	0	0.01	1	0	0	0.02
7	7	no	1	0	0	0	-	0	0.01	1	0	0	0.02
8	8	yes	5	0	0	0	-	0	0.02	8	0.02	6	0.06
8	8	no	8	1	1	0	-	0	0.04	8	0	0	0.05
9	12	yes	2	0	0	0	-	0	0.02	3	0	1	0.04
9	12	no	4	0	0	0	-	0	0.02	4	0	0	0.02
10	13	yes	11	0	0	0	-	0	0.08	14	0.05	4	0.14
10	13	no	13	0	0	0	-	0	0.06	13	0	0	0.08
11	17	yes	649	324	324	11	10	0.26	6.41	829	2.46	319	9.94
11	17	no	1063	531	531	17	10	0.33	7.91	1063	0	0	9.72
12	20	yes	17	1	1	0	-	0	0.22	24	0.25	11	0.53
12	20	no	848	414	414	0	-	1.75	5.69	848	0	0	8.97
13	26	yes	13	0	0	0	-	0	0.12	14	0.34	1	0.48
13	26	no	44	14	14	0	-	0	0.34	44	0	0	0.46
14	28	yes	77	27	27	0	-	0.05	1.82	121	0.85	86	2.91
14	28	no	314	145	145	0	-	0.15	3.57	314	0	0	4.71

Table 4.6: Branch-and-cut versus branch-and-bound for 2-$(v, 3, 1)$ packings

		clique separation algorithm	
V	MWV	general	specialized
$[0.1, 0.1]$	$[0.1, 0.3]$	run# 1	run# 2
$[0.1, 0.1]$	$[0.3, 0.3]$	run# 3	run# 4
$[0.3, 0.3]$	$[0.3, 0.3]$	run# 5	run# 6
$[0.3, 0.6]$	$[0.3, 0.6]$	run# 7	run# 8

Table 4.7: Parameter combination for several runs.

v	b	run#	BB	BT	LPT	LPs	ST	Cl	TotT
7	7	1..8	1	0	< 0.02	1	-	0	< 0.02
7	7	Cplex	0					0	0.01s
9	12	1..8	1	0	< 0.03	1	-	0	<0.04
9	12	Cplex	2					0	0.02
13	26	1	7	0	0.15	12	0.62	17	0.77
13	26	3	7	0	0.12	12	0.61	17	0.74
13	26	5	8	0	0.21	14	0.63	20	0.84
13	26	7	6	0	0.09	8	0.43	4	0.55
13	26	2	1	0	0.03	2	0.03	7	0.09
13	26	4	1	0	0.05	2	0.02	7	0.07
13	26	6	9	0	0.17	14	0.06	13	0.25
13	26	8	6	0	0.07	8	0.1	4	0.18
13	26	Cplex	17					0	0.34
15	35	1..8	1	0	0.05	1	0	0	< 0.08
15	35	Cplex	45					0	1.19
19	57	1	30	0	3.21	54	18.05	66	21.32
19	57	3	30	0	3.24	54	15.35	66	18.66
19	57	5	28	0	2.14	43	11.93	25	14.15
19	57	7	35	1	1.5	39	12.01	5	13.60
19	57	2	27	0	1.53	43	2.86	37	4.46
19	57	4	27	0	1.67	43	3.16	37	4.86
19	57	6	22	0	1.2	34	2.41	21	3.66
19	57	8	35	1	1.45	39	2.66	5	4.17
19	57	Cplex	52					0	6.52
21	70	1	42	0	5.57	63	29.13	51	34.78
21	70	3	42	0	5.67	63	29.3	51	35.05
21	70	5	39	0	2.38	49	24.07	16	26.57
21	70	7	58	8	4.13	63	29.34	7	33.73
21	70	2	34	0	5.2	56	6.59	59	11.87
21	70	4	34	0	5.24	56	6.59	59	11.95
21	70	6	39	0	2.72	49	5.39	15	8.18
21	70	8	58	8	3.86	63	6.17	7	10.29
21	70	Cplex	123					5	25.43

Table 4.8: Clique separation and parameter variations.

v	b	run#	BB	BT	LPT	LPs	ST	Cl	TotT
25	100	1	71	4	23.1	110	92.21	91	115.76
25	100	3	71	4	23.5	110	92.21	93	116.28
25	100	5	64	0	11.62	81	79.51	30	91.38
25	100	7	57	0	12.49	65	72.7	9	85.42
25	100	2	60	1	26.37	98	20.28	97	46.94
25	100	4	60	1	25.75	98	19.96	97	45.99
25	100	6	66	0	17.11	91	18.49	42	35.78
25	100	8	60	0	9.966	69	17.27	12	27.44
25	100	Cplex	97					1	63.83
27	117	1	75	3	50.46	124	156.84	121	207.89
27	117	3	74	3	42.84	121	153.92	118	197.35
27	117	5	102	18	74.5	144	146.66	81	222.79
27	117	7	76	4	32.89	89	132.96	16	166.52
27	117	2	71	2	41.69	112	33.66	109	75.86
27	117	4	71	2	43.31	112	33.75	109	77.55
27	117	6	99	14	48.23	137	32.58	63	82.29
27	117	8	76	4	32.61	89	29.89	16	63.09
27	117	Cplex	230					6	340.48
31	155	1	135	13	165.04	200	368.27	155	536.09
31	155	3	135	13	171.42	200	375.35	155	549.59
31	155	5	119	1	98.6	160	329.61	56	429.22
31	155	7	117	9	101.22	132	318.37	24	421.60
31	155	2	112	0	122.77	166	85.07	119	208.57
31	155	4	112	0	126.88	166	82.92	119	210.55
31	155	6	112	0	70.42	145	79.29	44	150.42
31	155	8	113	7	87.61	127	76.35	25	165.56
31	155	Cplex	2538					34	3879.00
33	176	1	141	11	381.24	215	526.76	181	911.19
33	176	3	141	11	382.21	215	537.29	181	922.63
33	176	5	219	52	768.50	295	508.96	170	1288.51
33	176	7	257	67	599.14	295	509	64	1122.38
33	176	2	118	5	255.11	182	122.8	152	379.71
33	176	4	118	5	252.65	182	133.36	152	387.85
33	176	6	123	0	106.33	155	112.80	42	220.02
33	176	8	174	26	309.52	198	114.61	53	430.36
33	176	Cplex	1267					14	2651.92

Table 4.9: Clique separation and parameter variations.

v	b	exist	BB	BT	LPT	LPs	ST	Cl	Sp	TotT
7	-	no	1	0	0	2	0.02	0	7	0.04
9	12	yes	1	0	0.03	1	0	0	0	0.04
13	-	no	2227	1113	187.37	5240	257.58	4771	16926	481.31
15	35	yes	26	7	19.91	61	7.12	40	144	27.73
19	57	yes	4820	2399	4909.88	12800	1925.24	12616	54257	(*) 7200

Table 4.10: 4-sparse 2-$(v, 3, 1)$ designs

v	b	exist	BB	BT	LPT	LPs	ST	Cl	Sp	TotT
7	-	no	1	0	0.01	2	0.02	0	8	0.03
9	-	no	1	0	0.06	2	0.07	0	36	0.13
13	-	no	31	15	12.96	122	13.77	40	962	30.06
15	35	no	4960	2478	3675.32	17386	2818.25	6357	182837	(*) 7200
19	57	yes	1180	582	4691.53	5007	2211.61	1757	58961	(*) 7200

Table 4.11: 5-sparse 2-$(v, 3, 1)$ designs

The problems with "(*) 7200" in the column TotT could not be solved within 2hs of CPU.

v	b	BB	BT	IE	IR	Mxl	IT	LPT	LPs	ST	Cl	TotT
8	14	1	0	0	0	-	0	0	1	0	0	0.01
10	30	1	0	0	0	-	0	0.07	1	0	0	0.07
14	91	1700	838	0	0	-	0	1974.75	1755	35.59	57	2488.7

Table 4.12: Steiner Quadruple Systems: 3-$(v, 4, 1)$ designs

v	b	exist.	BB	BT	IE	IR	Mxl	IT	LPT	LPs	ST	Cl	TotT
7	7	yes	1	0	0	0	-	0	0	1	0	0	0
9	-	no	1	0	0	0	-	0	0	1	0	0	0
13	26	yes	1	0	0	0	-	0	0.01	2	0	2	0.01
15	35	yes	2	0	0	0	-	0	0	3	0.04	3	0.04
19	57	yes	4	1	0	0	-	0	0.02	6	0.16	8	0.20
21	70	yes	4	1	0	0	-	0	0.02	8	0.38	16	0.44
25	100	yes	4	1	0	0	-	0	0.02	7	0.35	10	0.40
27	117	yes	9	3	0	0	-	0	0.04	13	0.52	9	0.61
31	155	yes	3	0	0	0	-	0	0.01	6	0.39	0	0.47

Table 4.13: Cyclic 2-$(v, 3, 1)$ designs

$t = 2$

	$k=3$				$k=4$			
v	Bl	C	D	S	Bl	C	D	S
8	1	8	8	8	1	2	2	4
9	1	9	12	12	0	0	3	4
10	1	10	13	13	0	0	5	7
11	1	11	17	18	0	0	6	8
12	2	16	20	20	1	3	6	9
13	2	26	26	26	1	13	13	13
14	1	14	28	28	1	14	14	14
15	3	35	35	35	1	15	15	15
16	2	32	37	37	1	16	20	20
17	2	34	44	45	1	17	20	21
18	3	42	48	48	1	18	22	22
19	3	57	57	57	1	19	25	28
20	2	40	60	60	2	25	30	30
21	4	70	70	70	1	21	31	31
22	3	66	73	73	1	22	37	38
23	3	69	83	84	1	23	40	40
24	4	80	88	88	2	30	42	42

Table 4.14: Cyclic 2-$(v, k, 1)$ packings

Bl: base blocks in the cyclic packing
C: size of maximal cyclic packing
D: size of maximal packing
S: Schönheim upper bound

$t = 3$

	$k=4$				$k=5$			
v	Bl	C	D	S	Bl	C	D	S
8	1	10	14	14	0	0	2	4
9	1	9	18	18	0	0	3	7
10	3	30	30	30	1	2	6	8
11	3	33	35	35	1	11	11	15
12	5	45	51	54	1	12	12	19
13	4	52	65	65	1	13	18	23
14	6	84	91	91	1	14	28	36
15	7	105	105	105	3	33	42	42
16	10	132	140	140	2	32	48	48
17	9	153	156	157	4	68	68	68
18	11	198	198	202	-	-	?	75
19	12	228	228	228	-	-	?	83
20	15	285	285	285	-	-	?	112
21	15	315	315	315	-	-	?	126

Table 4.15: Cyclic 3-$(v, k, 1)$ packings

- indicates our algorithm did not find the cyclic packings

? indicates that the regular packing number is unknown

$t = 4$

	$k = 5$				$k = 6$			
v	Bl	C	D	S	Bl	C	D	S
8	1	8	8	11	0	0	1	1
9	1	9	18	25	1	3	3	6
10	3	30	36	36	0	0	5	11
11	6	66	66	66	1	11	11	14
12	6	72	?	84	2	14	22	30
13	9	117	?	140	2	26	26	41
14	11	154	?	182	3	42	42	53

Table 4.16: Cyclic 4-$(v, k, 1)$ packings

$t = 5$

	$k = 6$				$k = 7$			
v	Bl	C	D	S	Bl	C	D	S
8	1	4	4	6	0	0	1	1
9	1	9	12	16	0	0	1	1
10	3	30	30	41	0	0	3	8
11	6	66	66	66	0	0	6	17
12	10	110	132	132	1	12	12	24
13	12	156	?	182	2	26	26	55
14	(*)	≥ 273	?	326	3	30	42	82
15	(*)	≥ 370	?	455	-	-	?	113

Table 4.17: Cyclic 5-$(v, k, 1)$ packings

-	indicates our algorithm did not find the cyclic packings
?	indicates that the regular packing number is unknown
(*)	indicates that a (not necessarily optimal) cyclic packing was found by the algorithm

v	C	D	Sch	BB	BT	LPT	LPs	ST	Cl	TotT
6	3	3	4	1	0	0	1	0.00	0	0.02
7	7	7	7	1	0	0	1	0.00	0	0.01
8	10	14	14	1	0	0	2	0.01	1	0.02
9	9	18	18	1	0	0	2	0.00	3	0.01
10	30	30	30	3	1	0.02	5	0.06	15	0.10
11	33	35	35	13	6	0.12	27	0.32	69	0.46
12	45	51	54	35	17	0.36	78	1.63	243	2.05
13	52	65	65	179	89	1.79	384	8.10	1074	10.03
14	84	91	91	254	126	3.68	549	19.32	1623	23.29
15	105	105	105	18	7	0.23	35	1.63	66	1.89
16	132	140	140	1321	660	32.21	2645	222.37	7680	257.28
17	153	156	157	7868	3933	257.49	16648	1446.2	66608	1716.41
18	198	198	202	6279	3139	230.44	12029	1759.78	33074	2002.21
19	228	228	228	2269	1131	96.89	467	411.81	15797	512.94
20	285	285	285	3597	1796	203.40	6611	1277.47	15262	1488.73
21	315	315	315	6916	3453	441.22	13433	1824.79	40252	2282.82

Table 4.18: Cyclic 3-$(v, 4, 1)$ packings

v	Base Blocks						
6	1 2 4 5						
7	1 2 3 5						
8	1 2 4 5						
9	1 2 4 6						
10	1 2 3 7	1 2 4 5	1 3 5 8				
11	1 2 3 5	1 2 6 8	1 2 7 10				
12	1 2 4 5	1 2 6 7	1 2 6 10	1 3 7 9	1 4 7 10		
13	1 2 4 9	1 2 5 6	1 2 8 12	1 3 5 9			
14	1 2 3 9	1 2 4 5	1 2 6 7	1 3 5 10	1 3 6 12	1 4 7 11	
15	1 2 3 12	1 2 4 8	1 2 6 10	1 2 7 13	1 2 9 14	1 3 5 8	1 3 10 13
16	1 2 3 8	1 2 4 11	1 2 5 14	1 2 6 13	1 2 9 10	1 2 12 15	1 3 6 12
	1 3 7 9	1 4 9 12	1 5 9 13				
17	1 2 3 7	1 2 5 11	1 2 8 13	1 2 9 15	1 2 10 14	1 2 12 16	1 3 6 10
	1 3 8 11	1 3 12 15					
18	1 2 3 5	1 2 6 7	1 2 8 16	1 2 9 10	1 2 13 17	1 3 6 16	1 3 7 11
	1 3 8 12	1 3 9 14	1 3 10 13	1 4 8 13			
19	1 2 3 18	1 2 4 14	1 2 6 11	1 2 7 9	1 2 8 16	1 2 10 12	1 2 13 15
	1 3 6 17	1 3 7 16	1 4 7 13	1 4 8 12	1 4 11 16		
20	1 2 3 8	1 2 4 5	1 2 6 9	1 2 10 15	1 2 11 17	1 2 12 14	1 2 13 16
	1 3 5 18	1 3 6 14	1 3 7 15	1 3 9 12	1 3 13 17	1 4 7 14	1 5 10 14
	1 6 11 16						
21	1 2 3 15	1 2 4 20	1 2 5 10	1 2 7 17	1 2 8 19	1 2 9 12	1 2 11 13
	1 2 16 18	1 3 5 18	1 3 9 17	1 3 10 14	1 4 7 13	1 4 8 17	1 4 12 18
	1 5 12 17						

Table 4.19: Cyclic 3-$(v, 4, 1)$ packings

v	C	D	Sch	BB	BT	LPT	LPs	ST	Cl	TotT
8	8	8	11	1	0	0	1	0	0	0
9	9	18	25	1	0	0	2	0.01	4	0.02
10	30	36	36	1	0	0	2	0.04	11	0.04
11	66	66	66	1	0	0	1	0	0	0.01
12	72	?	84	65	31	2.02	162	15.5	638	17.56
13	117	?	140	1639	819	82.45	4091	331.99	23919	416.61
14	154	?	182	5048	2523	395.94	11908	1685.53	59817	2090.79

Table 4.20: Cyclic 4-$(v, 5, 1)$ packings

v	Base Blocks					
8	1 2 3 4 6					
9	1 2 3 4 8					
10	1 2 3 4 9	1 2 4 6 8	1 2 5 6 9			
11	1 2 3 4 10	1 2 3 5 8	1 2 3 6 7	1 2 4 5 9	1 2 4 6 8	1 2 5 7 10
12	1 2 3 4 6	1 2 3 8 11	1 2 3 9 10	1 2 4 5 8	1 2 4 7 9	1 2 6 8 10
13	1 2 3 4 9	1 2 3 5 12	1 2 3 6 11	1 2 3 7 10	1 2 4 6 7	1 2 4 8 12
	1 2 5 8 9	1 2 6 8 10	1 3 5 8 11			
14	1 2 3 4 11	1 2 3 5 13	1 2 3 7 9	1 2 3 8 12	1 2 4 5 10	1 2 4 6 9
	1 2 4 7 8	1 2 5 9 11	1 2 6 11 13	1 2 8 10 13	1 3 5 9 12	

Table 4.21: Cyclic 4-$(v, 5, 1)$ packings

Chapter 5

Concluding Remarks and Open Problems

In this thesis, we have investigated the polyhedral structure of combinatorial design problems. This study has led to a new algorithmic technique for constructing various types of designs such as t-designs and packings, cyclic packings, and Steiner triple systems avoiding subconfigurations.

Extremal problems for set systems were a common theme: maximal intersecting families of sets (Chapter 3) and Erdös configurations (Section 2.3.3) are both defined based on extremal properties. Indeed, extremal problems are in the very nature of polyhedral theory and of the designs we dealt with. To study a polyhedron, one investigates its vertices and facets, which are its nonempty faces of smallest dimension and its proper faces of largest dimension. Maximal packings are the designs with most blocks such that each pairwise intersection of blocks does not exceed a given parameter t; maximal m-sparse packings are triple systems with most blocks avoiding the "concentration" of points in a few blocks. So, it is not surprising that many of the open problems we now describe are related to extremal set theory.

Polyhedral theory and designs

Problem 1. Use valid inequalities for the design polytopes to derive results for infinite families of designs.

In the algorithm of Chapter 4, we use these inequalities in order to determine maximal packing designs for specific parameters. So, the algorithm determines the packing number for fixed (t, v, k, λ). Can valid inequalities or facets (of the design polytope) be used to determine other properties, such as upper bounds on packing numbers, for an infinite family of parameters?

Algorithms involving set systems

An answer to the next question would be useful not only for problems discussed in this thesis, but also for designing other algorithms involving set systems.

Problem 2. Let $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_p$ be (nonisomorphic) families of k-sets of a v-set and let $\mathcal{B}$ be a family isomorphic to one of them.

1. Compute j such that family A_j is isomorphic to $\mathcal{B}$, in an efficient way. In particular, what is the minimum number of sets in $\mathcal{B}$ that have to be examined in order to determine A_j ?

2. If we know that A_j and $\mathcal{B}$ are isomorphic, what is the minimum number of sets of $\mathcal{B}$ to be examined in order to determine the permutation of $[1, v]$ that sends $\mathcal{B}$ to A_j?

We will illustrate these problems through an example. Suppose $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_p\} = MI^2(v, t+1, t)$, and $\mathcal{B}$ is an arbitrary family in $MI^2(v, t+1, t)$. Proposition 3.4.1 tells us three things: (1) $p = 2$, (2) by examining any 3 sets of $\mathcal{B}$ we can determine which of the two families $\mathcal{B}$ is isomorphic to, and (3) if $\mathcal{B}$ is isomorphic to $\binom{[1,t+2]}{t+1}$, then 2 sets are enough to determine the permutation that sends $\mathcal{B}$ to $\binom{[1,t+2]}{t+1}$. The clique separation algorithm in page 42 is based on these facts. A first step would be to extend these results to $(t+2)$-subsets of a v-set. In this case, $p = 17$ for $t \geq 2$, and the solution of these problems would give us an efficient clique separation algorithm for designs with $k = t+2$.

Problem 3. Design a branching scheme that uses the structure of the set systems for the branch-and-cut algorithm.

Given a subconfiguration corresponding to a node in the tree, the algorithm should branch according to the possible ways of extending this subconfiguration.

Problem 4. Find an efficient algorithm for enumeration of distinct (up to isomorphism) maximal intersecting set systems.

The naive algorithm generates all the set systems and then eliminates the isomorphic ones. An efficient algorithm should avoid the generation of isomorphic ones in the first place. General methods for this kind of generation are discussed in [74].

Maximal intersecting set systems

We pose a conjecture that generalizes some results from Chapter 3.

Conjecture 1. Let $k > t \geq 1$, $s \geq 2$. Then, there exist a $t_0 = t_0(k-t, s)$ and a $v_0 = v_0(k-t, s)$ such that any family in $MI^s(v, k, t)$ is an extension of a family in $MI^s(v_0, t_0 + (k-t), t_0)$, for all $t \geq t_0$, and $v \geq v_0 + (t - t_0)$. In particular, the number of families distinct up to isomorphism in $MI^s(v, k, t)$ is completely determined by $k - t$.

In this thesis, we prove this is true for $s = 2$ and $k - t = 1, 2$. For $s = 2$ and $k - t = 1$, Proposition 3.4.1 implies $t_0 = 1$ and $v_0 = 3$. For $s = 2$ and $k - t = 2$, Theorem 3.4.9 states that $t_0 = 2$ and $v_0 = 8$. In the first case, this means $MI^2(3, 2, 1)$ completely determines $MI^2(v, t+1, t)$ for any $t \geq 1$ and $v \geq t+3$, and in the second case $MI^2(8, 4, 2)$ completely determines all $MI^2(v, t+2, t)$ for any $t \geq 2$ and $v \geq t+6$. Is it true that the same happens for other values of $(k-t)$?

The proof of this conjecture would imply that the classification problem (Problem 2 on page 58) for maximal s-wise t-intersecting systems of k-sets of a v-set depends only on s and $k - t$. We have already shown it does not depend on v (Corollary 3.2.10).

The following conjecture concerns kernels and essential sets of intersecting families (see Chapter 3).

Conjecture 2. Let $s \geq 2$ and $t < k$ and, for any $v > k$, let $e(v) = \max_{\mathcal{A} \in MI^s(v,k,t)} |E(\mathcal{A})|$. Then, if $\overline{v}$ is the smallest integer such that $e(\overline{v}) < \overline{v}$, then $n(t, s, k) = e(\overline{v})$.

This conjecture implies that whenever the maximum size among all kernels of families in $MI^s(v, k, t)$ is strictly smaller than v, no other family with $v' \geq v$ can have a kernel of size v. If true, we can design an algorithm to determine $n(t, s, k)$, that is able to halt whenever such a $\overline{v}$ is found, and so has to examine up to $n(t, s, k)$ different v's. This conjecture is true for $s = 2$ and $k - t = 1, 2$ (see Table 3.1).

Other algorithmic problems

There are other natural extensions of our results and algorithms that emerge from this thesis. For example, other classes of facets could be investigated, the branch-and-cut algorithm could be extended to deal with other kinds of designs (such as coverings, designs with higher λ, and admitting other automorphism groups). In addition, it would be interesting to investigate whether the combination of techniques such as tactical decompositions (see [71]) with the polyhedral approach leads to more efficient algorithms.

Bibliography

[1] R.J.R. Abel and M. Greig. *BIBDs with small block size*, chapter I.2, pages 41–47. In Colbourn and Dinitz [24], 1996.

[2] R. Ahlswede and L.H. Khachatrian. The complete nontrivial-intersection theorem for systems of finite sets. *J. Combin. Theory. Ser. A*, 76:121–138, 1996.

[3] R. Ahlswede and L.H. Khachatrian. The complete intersection theorem for systems of finite sets. *Europ. J. Combin.*, 18:127–136, 1997.

[4] N. Alon and Z. Füredi. On the kernel of intersecting families. *Graphs Combin.*, 3:91–94, 1987.

[5] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Finding cuts in the TSP. Preliminary report, 1994.

[6] A. Bachem and M. Grötschel. New aspects of polyhedral theory. In *Modern Applied Mathematics, Optimization and Operations Research*, pages 51–106. North-Holland Publishing Company, 1982.

[7] E. Balas and S.M. Ng. On the set covering polytope: I. all the facets with coefficients in $\{0, 1, 2\}$. *Math. Programming*, 43:57–69, 1989.

[8] E. Balas and M.W. Padberg. On the set covering problem. *Oper. Res.*, 20:1153–1161, 1972.

[9] E. Balas and M.W. Padberg. Set partitioning: a survey. *SIAM Review*, 18:710–760, 1976.

[10] E. Balas and E. Zemel. Critical cutsets of graphs and canonical facets of set-packing polytopes. *Math. Oper. Res.*, 2:15–19, 1977.

[11] F. Barahona and A.R. Mahjoub. Compositions of graphs and polyhedra II: Stable sets. *SIAM J. Disc. Math.*, 7:359–371, 1994.

[12] A. Betten, E. Laue, and A. Wassermann. New t-designs and large sets of t-designs. *Discrete Math.*, 1997. to appear.

[13] R. Borndörfer. *Aspects of Set Packing, Partitioning and Covering.* PhD thesis, Universität Berlin, 1997.

[14] A.E. Brouwer. Steiner triple systems without forbidden configurations. Technical Report ZW104/77, Mathematisch Centrum Amsterdam, 1977.

[15] M. Całczyńska-Karłowicz. Theorem on families of finite sets. *Bull. Acad. Polon. Sci., Ser. Math. Astron. Phys.*, 12:87–89, 1964.

[16] A. Caprara and M. Fischetti. *Branch-and-cut algorithms*, chapter 4. In Dell'Amico et al. [31], 1997.

[17] S. Ceria, P. Nobili, and A. Sassano. *Set Covering Problem*, chapter 23. In Dell'Amico et al. [31], 1997.

[18] Y.M. Chee, C.J. Colbourn, and A.C.H. Ling. Asymptotically optimal erasure-resilient codes for large disk arrays. Preprint (42p.).

[19] E. Cheng and W.H. Cunningham. Wheel inequalities for stable set polytopes. *Math. Programming*, 77:389–421, 1997.

[20] T. Christof and A. Loebel. Porta software. http://www.iwr.uni-heidelberg.de/iwr/comopt /soft/PORTA/readme.html.

[21] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Math.*, 4:305–337, 1973.

[22] V. Chvátal. On certain polytopes associated with graphs. *J. Combin. Theory. Ser. B*, 18:138–154, 1975.

[23] D.M. Cohen, S.R. Dalal, J. Parelius, and G.C. Patton. The combinatorial design approach to automatic test generation. *IEEE Software*, 13:83–88, 1996.

[24] C.J. Colbourn and J.H. Dinitz, editors. *The CRC handbook of combinatorial designs.* CRC Press, 1996.

[25] C.J. Colbourn and R. Mathon. *Steiner systems*, chapter I.4, pages 66–75. In Colbourn and Dinitz [24], 1996.

[26] C.J. Colbourn, E. Mendelsohn, A. Rosa, and J. Širáň. Anti-mitre Steiner triple systems. *Graphs Combin.*, 10:215–224, 1994.

[27] C.J. Colbourn and P.C. van Oorschot. Applications of combinatorial designs in computer science. *ACM Computing Surveys*, 21:223–250, 1989.

[28] G. Cornuéjols and A. Sassano. On the 0,1 facets of the set covering polytope. *Math. Programming*, 43:45–55, 1989.

[29] *Using the CPLEX Callable Library, version 4.0*, 1995. http://www.cplex.com.

[30] H. Crowder, E.L. Johnson, and M. Padberg. Solving large-scale zero-one linear programming problems. *Oper. Res.*, 31:803–834, 1983.

[31] M. Dell'Amico, F. Maffioli, and S. Martello, editors. *Annotated Bibliographies in Combinatorial Optimization.* John Wiley & Sons, 1997.

[32] A. Ehrenfeucht and J. Mycielski. Interpolation of functions over a measure space and conjectures about memory. *J. Approx. Theory*, 9:218–236, 1973.

[33] P. Erdös, C. Ko, and R. Rado. Intersection theorems for systems of finite sets. *Quart. J. Math. Oxford*, 12:313–320, 1961.

[34] P. Erdös and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In A. Hajnal, R. Rado, and V.T. Sós, editors, *Infinite and finite sets*, pages 609–627. Keszthely, Hungary, 1973.

[35] R. Euler, M. Jünger, and G. Reinelt. Generalizations of cliques, odd holes and anticycles and their relation to independence system polyhedra. *Math. Oper. Res.*, 12:451–462, 1987.

[36] P. Frankl. The Erdös-Ko-Rado theorem is true for $n = ckt$. *Coll. Soc. Math. J. Bolyai*, 18:365–375, 1978.

[37] P. Frankl. On intersecting families of finite sets. *J. Combin. Theory. Ser. A*, 24:146–161, 1978.

[38] P. Frankl. Multiple-intersecting families. *J. Combin. Theory. Ser. B*, 53:195–234, 1991.

[39] P. Frankl and Z. Füredi. Finite projective spaces and intersecting hypergraphs. *Combinatorica*, 6:335–354, 1986.

[40] D.R. Fulkerson. Blocking and anti-blocking pairs of polyhedra. *Math. Programming*, 1:168–194, 1971.

[41] Z. Füredi. Geometrical solution of an intersection problem for two hypergraphs. *Europ. J. Combin.*, 5:133–136, 1984.

[42] Z. Füredi. Matchings and cover in hypergraphs. *Graphs Combin.*, 4:115–206, 1988.

[43] P.B. Gibbons. *Computational methods in design theory*, chapter VI.9, pages 718–740. In Colbourn and Dinitz [24], 1996.

[44] C.D. Godsil. Linear algebra and designs. Manuscript (229 p.), 1995.

[45] M.J. Grannell and T.S. Griggs. Configurations in Steiner triple systems. Preprint (33p.).

[46] M.J. Grannell, T.S. Griggs, and E. Mendelsohn. A small basis for four-line configurations in Steiner triple systems. *J. Combin. Des.*, 3(1):51–59, 1995.

[47] M.J. Grannell, T.S. Griggs, and J.S. Phelan. A new look at an old construction for Steiner triple systems. *Ars Combin.*, 25A:55–60, 1988.

[48] T.S. Griggs and J.P. Murphy. 101 anti-Pasch Steiner triple systems of order 19. *J. Combin. Math. Combin. Comput.*, 13:129–141, 1993.

[49] A.J.W. Hilton and E.C. Milner. Some intersection theorems for systems of finite sets. *Quart. J. Math. Oxford*, 18:369–384, 1967.

[50] K. Hoffman and M. Padberg. Solving airline crew-scheduling problems by branch-and-cut. *Manag. Sci.*, 39:657–682, 1993.

[51] J. Kahn and P. Seymour. Unpublished.

[52] W. Kocay. *On writing isomorphism programs*, pages 135–175. In Wallis [102], 1996.

[53] Y. Kohayakawa. A note on kernels of intersecting families. *Europ. J. Combin.*, 11:155–164, 1990.

[54] E.S. Kramer. Some t-designs for $t \geq 4$ and $t = 17, 18$. *Congr. Numer.*, 14:443–460, 1975.

[55] E.S. Kramer, D.W. Leavitt, and S.S. Magliveras. Construction procedures for t-designs and the existence of new simple 6-designs. *Ann. Disc. Math.*, 26:247–274, 1985.

[56] E.S. Kramer and D.M. Mesner. t-designs on hypergraphs. *Discrete Math.*, 15:263–296, 1976.

[57] D.L. Kreher. *t-designs, $t \geq 3$*, chapter I.3, pages 47–66. In Colbourn and Dinitz [24], 1996.

[58] D.L. Kreher and S.P. Radziszowski. The existence of simple 6–$(14, 7, 4)$ designs. *J. Combin. Theory. Ser. A*, 43:237–243, 1986.

[59] D.L. Kreher and S.P. Radziszowski. Finding simple t-designs by using basis reduction. *Congr. Numer.*, 55:235–244, 1986.

[60] D.L. Kreher and S.P. Radziszowski. New t-designs found by basis reduction. *Congr. Numer.*, 59:155–164, 1987.

[61] C.W.H. Lam, L. Thiel, and S. Swiercz. The nonexistence of finite projective planes of order 10. *Canad. J. Math.*, 41:1117–1123, 1989.

[62] M. Laurent. A generalization of antiwebs to independence systems and their canonical facets. *Math. Programming*, 45:97–108, 1989.

[63] H. Lefmann, K.T. Phelps, and V. Rödl. Extremal problems for triple systems. *J. Combin. Des.*, 1:379–394, 1993.

[64] A.K. Lenstra, H.W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.

[65] J.K. Lenstra and A.H.G. Rinnooy Kan. Complexity of packing, covering and partitioning. In A. Schrijver, editor, *Packing and Covering in Combinatorics*, pages 275–291. Mathematical Centre Trats 106, 1979.

[66] A.C.H. Ling. A direct product construction for 5-sparse triple systems. *J. Combin. Des.* to appear.

[67] A.C.H. Ling, C.J. Colbourn, M.J. Grannell, and T.S. Griggs. Construction techniques for anti-Pasch Steiner triple systems. Preprint (22p.), 1997.

[68] L. Lovász and A. Schrijver. Cones of matrices and set-functions and $0-1$ optimization. *SIAM J. Optimization*, 1:166–190, 1991.

[69] S.S. Magliveras and D.W. Leavitt. Designs from $P\Gamma L_2(32)$. In *Computational Group Theory, Proceedings of the London Mathematical Society Symposium on Computational Group Theory*, pages 337–352. Academic Press, 1984.

[70] R. Mathon. Unpublished.

[71] R. Mathon. *Computational methods in design theory*, pages 29–48. In Wallis [102], 1996.

[72] R. Mathon and A. Rosa. 2–(v, k, λ) *designs of small order*, chapter I.1, pages 3–41. In Colbourn and Dinitz [24], 1996.

[73] B.D. McKay. Nauty user's guide (version 1.5). Technical Report TR-CS-90-02, Dept. Computer Science, Austral. Nat. Univ., 1990. http://cs.anu.edu.au/people/bdm/nauty/.

[74] B.D. McKay. Isomorph-free exhaustive generation. *J.Algorithms*, 26:306–324, 1998.

[75] W.H. Mills and R.C. Mullin. Coverings and packings. In J.H. Dinitz and D.R. Stinson, editors, *Contemporary Design Theory: a collection of surveys*, pages 371–399. Wiley, 1992.

[76] L. Moura. *Polyhedral methods in design theory*, pages 227–254. In Wallis [102], 1996.

[77] L. Moura. Maximal s-wise t-intersecting families of sets: kernels, generating sets, and enumeration. *J. Combin. Theory. Ser. A*, to appear (accepted in June 1998), 22 pages.

[78] G.L. Nemhauser and G. Sigismondi. A strong cutting plane/branch-and-bound algorithm for node packing. *J. Opl. Res. Soc.*, 43:443–457, 1992.

[79] G.L. Nemhauser and L.E. Trotter Jr. Properties of vertex packing and independence system polyhedra. *Math. Programming*, 6:48–61, 1974.

[80] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, 1988.

[81] P. Nobili and A. Sassano. Facets and lifting procedures for the set covering polytope. *Math. Programming*, 45:111–137, 1989.

[82] M.W. Padberg. On the facial structure of set packing polyhedra. *Math. Programming*, 5:199–215, 1973.

[83] M.W. Padberg and G. Rinaldi. Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Oper. Res. Letters*, 6:1–7, 1987.

[84] M.W. Padberg and G. Rinaldi. A branch-and-cut algorithm for the solution of large-scale traveling salesman problems. *SIAM Review*, 33:60–100, 1991.

[85] K. Phelps, A. Rosa, and E. Mendelsohn. Cyclic Steiner triple systems with cyclic subsystems. *Europ. J. Combin.*, 10:363–367, 1989.

[86] W.R. Pulleyblank. Polyhedral combinatorics. In N.L. Nemhauser et al., editor, *Handbooks in OR & MS*, pages 371–446. North-Holland, 1989.

[87] D.K. Ray-Chaudhuri and R.M. Wilson. On t-designs. *Osaka J. Math.*, 12:737–744, 1975.

[88] R.M. Robinson. The structure of certain triple systems. *Math. Comput.*, 29:223–241, 1975.

[89] A. Sassano. On the facial structure of the set covering polytope. *Math. Programming*, 44:181–202, 1989.

[90] B. Schmalz. The t-designs with prescribed automorphism group, new simple 6-designs. *J. Combin. Des.*, 1:125–170, 1993.

[91] A. Schrijver. On cutting planes. *Ann. Disc. Math.*, 9:291–296, 1980.

[92] Y. Sekiguchi. A note on node packing polytopes on hypergraphs. *Oper. Res. Letters*, 2:243–247, 1983.

[93] N. Stefan and K. Mehlhorn. LEDA-Manual version 3.0. Technical Report MPI-I-93109, Max-Planck-Institute für Informatik, 1993. http://www.mpi-sb.mpg.de/LEDA/leda.html.

[94] B. Stevens, L. Moura, and E. Mendelsohn. Lower bounds for transversal covers. *Des. Codes Cryptogr.*, 15(3):279–300, 1998.

[95] D.R. Stinson. *Coverings*, chapter IV.8, pages 260–265. In Colbourn and Dinitz [24], 1996.

[96] D.R. Stinson. *Packings*, chapter IV.33, pages 409–413. In Colbourn and Dinitz [24], 1996.

[97] D.R. Stinson and Y.J. Wei. Some results on quadrilaterals in Steiner triple systems. *Discrete Math.*, 105:207–219, 1992.

[98] A.P. Street and D.J. Street. *Combinatorics of Experimental Designs*. Oxford University Press, 1987.

[99] L. Teirlinck. Non-trivial t-designs without repeated blocks exist for all t. *Discrete Math.*, 65:301–311, 1987.

[100] L.E. Trotter Jr. A class of facet producing graphs for vertex packing polyhedra. *Discrete Math.*, 12:373–388, 1975.

[101] Zs. Tuza. Critical hypergraphs and intersecting set-pair systems. *J. Combin. Theory. Ser. B*, 39:134–145, 1985.

[102] W.D. Wallis, editor. *Computational and Constructive Design Theory*. Kluwer Academic Publishers, 1996.

[103] D. Wengrzik. Schnittebenenverfahren für blockdesign-probleme. Master's thesis, Universität Berlin, 1995.

[104] R.M. Wilson. The necessary conditions for t-designs are sufficient for something. *Util. Math.*, 4:207–215, 1973.

[105] R.M. Wilson. The exact bound on the Erdös-Ko-Rado theorem. *Combinatorica*, 4:247–257, 1984.