A NEW POLYHEDRAL APPROACH TO COMBINATORIAL DESIGNS

A Dissertation

by

IVETTE ARAMBULA MERCADO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2004

Major Subject: Industrial Engineering

A NEW POLYHEDRAL APPROACH TO COMBINATORIAL DESIGNS

A Dissertation

by

IVETTE ARAMBULA MERCADO

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Illya V. Hicks
(Chair of Committee)

Richard M. Feldman
(Member)

Jianer Chen
(Member)

V. Jorge Leon
(Member)

Brett A. Peters
(Head of Department)

May 2004

Major Subject: Industrial Engineering

ABSTRACT

A New Polyhedral Approach to Combinatorial Designs. (May 2004)

Ivette Arámbula Mercado, B.S., Tecnológico de Monterrey (ITESM);

M.S., Tecnológico de Monterrey (ITESM)

Chair of Advisory Committee: Dr. Illya V. Hicks

We consider combinatorial t-design problems as discrete optimization problems. Our motivation is that only a few studies have been done on the use of exact optimization techniques in designs, and that classical methods in design theory have still left many open existence questions. Roughly defined, t-designs are pairs of discrete sets that are related following some strict properties of size, balance, and replication. These highly structured relationships provide optimal solutions to a variety of problems in computer science like error-correcting codes, secure communications, network interconnection, design of hardware; and are applicable to other areas like statistics, scheduling, games, among others.

We give a new approach to combinatorial t-designs that is useful in constructing t-designs by polyhedral methods. The first contribution of our work is a new result of equivalence of t-design problems with a graph theory problem. This equivalence leads to a novel integer programming formulation for t-designs, which we call GDP. We analyze the polyhedral properties of GDP and conclude, among other results, the associated polyhedron dimension. We generate new classes of valid inequalities to aim at approximating this integer program by a linear program that has the same optimal solution. Some new classes of valid inequalities are generated as Chvátal-Gomory cuts, other classes are generated by graph complements and combinatorial arguments, and others are generated by the use of incidence substructures in a t-design. In particular, we found a class of valid inequalities that we call stable-set class that represents an alternative graph equivalence for the problem of finding a t-design. We analyze and give results on the strength of these new classes of valid inequalities.

We propose a separation problem and give its integer programming formulation as a maximum (or minimum) edge-weight biclique subgraph problem. We implement a pure cutting-plane algorithm using one of the stronger classes of valid inequalities derived. Several instances of t-designs were solved efficiently by this algorithm at the root node of the search tree. Also, we implement a branch-and-cut algorithm and solve several instances of 2-designs trying different base formulations. Computational results are included.

To my three beloved children

César Abiel, Juliette, and Alejandro

ACKNOWLEDGMENTS

I am truly indebted to my mentor, Dr. Richard M. Feldman, for being the professor that helped me the most in this academic endeavor from the beginning. He was always there to guide, give a wise word of advice, support me in prayer, and encourage me in tough times. He has been a blessing for me and my family. My admiration, respect, and gratitude go to him, for the indelible seal he imprinted in my professional life.

My appreciation also goes to my advisor, Dr. Illya Hicks, for introducing me to these interesting, but challenging problems in this dissertation, for his patience, and for the assistantship support in the last months of my studies. Special thanks also to Dr. Jianer Chen and Dr. Jorge Leon, for their willingness to give from their valuable time and expertise by serving as members of my committee.

My gratitude goes also to Judy Meeks, the Graduate Supervisor at the Department of Industrial Engineering. She was always friendly, willing to help, and assisted me many times during my doctoral studies. Also from the Department, I thank the Computer Systems staff, Dennis Allen and Mark Henry, for facilitating my use of the computer equipment and resolving technical problems during the time I worked on this research.

I thank the assistantship support from the Department of Industrial Engineering at Texas A&M University. It was my pleasure to work as teaching assistant for professors Dr. Richard Feldman, Dr. Robert Shannon, and Dr. Bryan Deuermeyer. I thank Dr. Wilbert Wilhelm for the opportunity to work with him as research assistant on a project for the Texas Engineering Experiment Station. Also, I thank the earlier sponsorship of ITESM, in particular to the former Dean, Juan Manuel Silva, and to Tomás Sánchez, Head of the Mathematics Department.

I express my warmest thanks and affection to my husband, César, for his continued patience, love and support during the years of my doctoral studies. I thank my parents, Martín and Lilia, for all their love and encouragement, and my sisters Lizette, Anette and Edith for all their support and help. But foremost, I am grateful to my heavenly Father for His abounding blessings and faithfulness. Yours, *O LORD*, is the greatness and the power and the glory and the majesty and the splendor, for everything in heaven and earth is yours [1 Chronicles 29:11].

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

TABLE Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

This dissertation studies the application of polyhedral theory and algorithms to combinatorial designs. We consider combinatorial t-design problems as discrete optimization problems. Integer and combinatorial optimization deals with problems of maximizing or minimizing a function of many variables subject to some constraints and subject to integrality conditions in some or all of the variables. A great number and variety of practical problems can be represented by discrete optimization models.

A Combinatorial Optimization problem is the problem of, given a ground finite set N and a family $\mathcal{F}$ of feasible subsets of N, finding a subset from $\mathcal{F}$ that either maximizes or minimizes some given criteria. In principle, this problem can be solved by enumeration, but it turns out that in practice the number of possible solutions is astronomically large. The theory of combinatorial optimization focuses in devising algorithms and methods to solve these problems in a more efficient way.

Roughly speaking, the representation of a problem by means of inequalities and equalities, jointly with an objective function and conditions on the variables is called a *formulation*. There may exist several formulations for the same problem. When the variables are not restricted to integer values, the problem is called *linear program*. Linear programs can be solved efficiently, and in essence any formulation leads to an optimal solution. The case when the variables have integrality con-

---

The journal model is *INFORMS Journal on Computing*.

ditions, called *integer programs*, is totally different. From Nemhauser and Wolsey [49] we quote: in integer programming, formulating a "good" model is of crucial importance to solving the model.

The main question in t-designs given admissible parameters is the problem of existence. Design theory provides necessary conditions for the existence of t-designs, but in general they are not sufficient. The problem seems intrinsically difficult. Cameron [11] points out that there is no hope in deciding which t-designs exist. Moura [47] indicates that it is not known if the set-packing problem applied to finding a t-design is $\mathcal{NP}$-complete. Furino [24] states that a computational search to try to find a new combinatorial design can be an intimidating task. There are many open existence problems for block designs and t-designs.

Only recent polyhedral studies applied to the specific case of t-designs have been pursued. We found that basically two formulations for the problem have been studied in the literature. Some studies are for particular cases of the problem, others for relaxed versions of the problem. Several studies include cutting plane algorithms and branch-and-cut applications to finding a t-design.

The research question of this dissertation is to study the existent formulations for t-designs, and to investigate if there is a possible alternative way to formulate the problem. This question is motivated first by the above idea on how important a "good" formulation is, motivated also by the fact that design theory has still left many open existence questions, and finally motivated by the fact that little research has been pursued for establishing a link between polyhedral theory and combinatorial t-designs.

As a result of our study, we were able to derive some theoretical polyhedral results for the problem, and to apply these results in a computational implementation of a branch-and-cut algorithm. One of the main contributions of this work is a proof that t-design problem has an equivalent graph problem. This equivalence resulted in a novel integer programming formulation for t-designs, which we call GDP. We give a study of the polyhedral properties of the polytope associated with GDP. We derive classes of valid inequalities by applying known polyhedral theory methods like Chvátal-Gomory cuts and cutting-plane proofs. We also were able to derive stronger classes of valid inequalities and to give a strength proof using t-design properties.

For the branch-and-cut algorithm, we propose a separation problem and a formulation for it. We compared computationally the strong bounds derived analytically and confirmed they were exact as shown. The computational tests on the separation problem show that, for equivalent valid inequalities, some classes performed consistently better than others. Finally, the effect of different choices of base formulations and warm start considerations are evaluated and computational results included.

In the first part of Chapter II we give an introduction to t-designs. We define the problem, give some applications to highlight their importance, and mention some classical construction methods to contrast with the polyhedral approach we will present in this work. In the second part of this chapter we give a brief review of polyhedral theory and algorithms. We include integrality of polyhedra, complexity and problem reductions, the relationship between separation and optimization, and the general branch-and-cut algorithm.

In Chapter III, we present the current research status concerning polyhedral approaches to combinatorial designs. We include existent integer programming formulations found in the literature, describe some recent polyhedral applications, include tables of some of the open problems with special emphasis in 2-designs, and include other recent computational approaches found.

In Chapter IV, we present the new problem equivalence result, and the consequent integer programming formulation GDP that we propose and study. We give also a polyhedral analysis on GDP that includes the dimension of the polyhedron associated with it. We further study GDP with the purpose of deriving new classes of valid inequalities. In Chapter V, we first derive a lower bound for the original biclique inequalities in GDP. Then apply Chvátal-Gomory cuts to derive other classes of valid inequalities.

In Chapter VI, we derive other classes of valid inequalities by graph complementing and supplementing incidence structures. We prove the strength of the LP relaxation bounds for these classes. We also derive interesting valid inequalities from substructures on t-design, that turned out to be strong and useful in the computational implementation. Finally, in Chapter VIII, we describe the separation problem we propose, a branch-and-cut implementation, and include computational results.

CHAPTER II

BACKGROUND

For the reader not familiar with combinatorial designs, in the first part of this chapter we give a brief introduction to the topic. We include basic definitions and some applications to highlight the importance of designs and the diversity of areas in which these problems can be useful. We also mention the classical methods in constructing combinatorial designs to contrast with the polyhedral approach that we will present in this work. A comprehensive source of information in combinatorial designs is a handbook edited by Colbourn and Dinitz [13]. An extensive treatment in design theory can be found in the encyclopedic work of Beth, Jungnickel, and Lenz [8].

The second part of this chapter gives an overview of polyhedral theory and methods. Excellent sources on this topic are the books by Cook et al. [17]; Nemhauser and Wolsey [49]; Schrijver [51]; Wolsey [63]. An accessible introduction to the branch-and-cut method can be found in Wolsey [63]. A comprehensive source of information in combinatorics is the handbook by Graham, Grötschel, and Lovász [27]. A good source of updated results and algorithms in combinatorial optimization can be found in the recent work of Korte and Vyjen [36].

II.1. Combinatorial t-designs

II.1.1. Definitions

The most basic notion in finite geometry is that of an incidence structure [8]. The idea is that two objects from different classes may be related with each other. The only requirement is that the classes do not overlap.

Definition An *incidence structure* is a triple $D = (X, \mathcal{B}, I)$ where X and $\mathcal{B}$ are disjoint sets and I is a binary relation between X and $\mathcal{B}$, i.e. $I \subseteq X \times \mathcal{B}$. The elements of X are called *points*, those of $\mathcal{B}$ are called *blocks*, and those of I are called *flags*.

An incidence structure is *finite* if both X and $\mathcal{B}$ are finite. In this work, we will deal only with finite incidence structures. Also, we focus our attention to incidence structures whose set $\mathcal{B}$ is a set of sets, that is, *set systems* in mathematical terms. More specific, each element of $\mathcal{B}$ is a subset of points of X (note that X and $\mathcal{B}$ are still disjoint). For the rest of this work, the incidence structures studied will be only finite incidence structure with blocks as subsets of points.

In an incidence structure, repeated blocks are allowed so that different blocks may correspond to the same subset of X. Incidence structures with no repeated blocks are called *simple*. Given an incidence structure $D = (X, \mathcal{B}, I)$, and given a point $p \in X$, the number of blocks from $\mathcal{B}$ that contain p is called the *point replication* of p. Given a block $B \in \mathcal{B}$, the number of points that B is comprised of is called the *block size* of B.

An incidence structure is said to be *uniform* with block size k if all blocks contain exactly k points. Uniform incidence structures are also studied under the name of *k-hypergraphs*. A 2-hypergraph is simply a graph. A uniform incidence structure is called *complete* if each subset of size k of the point set appears as a block, otherwise is called *incomplete*.

A set of say, t, elements is called a t-set. For positive integers t and λ, $t < |X|$, a finite incidence structure $D = (X, \mathcal{B}, I)$ is called *t-balanced* if there exists an integer $\lambda \geq 1$ such that every t-set of X appears in exactly λ blocks from $\mathcal{B}$. The number λ

is called the *index* of the balanced incidence structure.

Combinatorial t-designs, the main subject of our study, are finite incidence structures with certain properties. The definitions follows,

Definition A *t-design* on v points with parameters k and λ, denoted *t-(v,k,λ)* is an incidence structure $D = (X, \mathcal{B}, I)$ that satisfies the following conditions:

1. $|X| = v$;
2. D is uniform with block size k;
3. D is t-balanced with index λ.

The conditions in the definition of a t-design imply that the point replication is the same for all points. For historical reasons, it is customary that the point replication of a t-design is denoted r, and the number of blocks is denoted b. That is, b denotes the cardinality of $\mathcal{B}$.

The parameters of a t-design v, k and λ are natural numbers, which satisfy $t < k < v$, $\lambda > 0$. To define an instance of a t-design, only those parameters t-(v,k,λ) are needed because b and r can be computed from them using the following,

Theorem 1 (see [8, 11, 13]) *If $(X,\mathcal{B})$ is a t-(v,k,λ) design and S is any s-element subset of X, with $0 \leq s \leq t$, then the number of blocks containing S is given by*

$$\lambda_s = |\{B \in \mathcal{B} : S \subseteq B\}| = \frac{\lambda\binom{v-s}{t-s}}{\binom{k-s}{t-s}}, \qquad 0 \leq s \leq t \tag{2.1}$$

In particular, the number of blocks $b = \lambda_0$, the point replication $r = \lambda_1$, and the index $\lambda = \lambda_t$. Since λ_s in (2.1) needs to be an integer, only the values of v, k and λ that make λ_s integer for all $0 \leq s \leq t$ are *admissible parameters* for a t-design. A consequence of Theorem 1 is the following,

Corollary 2 (see [55]) *A t-design is also a s-design for $0 \leq s \leq t$.*

A t-design with $v = b$ and $t \geq 2$ is called a *symmetric* design (see van Trung [58] for an introduction to symmetric designs). A t-design with $3 \leq k < v$ is called *nontrivial*. Some special cases of t-(v, k, λ) designs have special names in the literature, Table 1 includes a few.

Although our theoretical study is in general for t-designs, our computational study is focused on the special case of t-designs when $t = 2$, that is, block designs. Here we include a somehow simpler definition of this type of problems.

Definition A *block design* on v points and b blocks is a pair $(X, \mathcal{B})$ where X is a set whose elements are called *points* and $\mathcal{B}$ is a collection of k-subsets of X called *blocks*, such that each element of v is contained in exactly r blocks and any 2-subset of X is contained in exactly λ blocks.

A simplified version of equation (2.1) for the case $t = 2$ for computing the number of blocks and the point replication parameters of a block design is as follows.

Theorem 3 (see [8]) *For a block design 2-(v, k, λ),*

$$r = \lambda(v-1)/(k-1) \tag{2.2}$$

$$b = \lambda v(v-1)/k(k-1) \tag{2.3}$$

The fundamental question in the study of designs is to establish necessary and sufficient condition for their existence. Here we include some theoretical results for block designs. As r and b must be natural numbers, the admissible parameters from Theorem 1 for the special case $t = 2$ lead to the the following,

Table 1. Some special cases of t-designs, their names and notation

Special Case of t-(v,k,λ)	Literature Name	Alternative Notation
$t = 2$	Block Design or Balanced Incomplete Block Design (BIBD)	2-(v,k,λ) or $S_\lambda(2,k,v)$ or (v,k,λ)-BIBD
$t = 2$, $k = 3$	Triple System	TS(v,λ)
$t = 2$, $k = 3$, $\lambda = 1$	Steiner Triple System	STS(v)
$t = 3$, $k = 4$, $\lambda = 1$	Steiner Quadruple System	SQS(v)
$\lambda = 1$	Steiner System	S(t,k,v)
$t = 2$, $k = q$, $v = q^n$, $\lambda = 1$, for a prime power q, $n \geq 2$	Affine Geometries (affine planes for n=2)	AG(n,q)
$t = 2$, $k = q+1$, $v = q^n + \cdots + q + 1$, $\lambda = 1$, for a prime power q, $n \geq 2$	Projective Geometries (projective planes for n=2)	PG(n,q)
$t = 2$, $v = 4n-1$, $k = 2n-1$ and $\lambda = n-1$ for natural number $n \geq 2$	Hadamard Design	HD(n)
$t = 2$, and $\mathcal{B}$ can be partitioned into parallel classes, each of which partitions X	Resolved Balanced Incomplete Block Design (RBIBD)	(v,k,λ)-RBIBD

Corollary 4 (see [8]) *The necessary conditions for the existence of a block design 2-*(v,k,λ) *are*

$$\lambda(v-1) \equiv 0 \;\; mod \;\; (k-1)$$

$$\lambda v(v-1) \equiv 0 \;\; mod \;\; k(k-1)$$

Another necessary condition is given by the next theorem,

Theorem 5 (The Fisher inequality) *For an arbitrary 2-*(v,k,λ) *design with* $k < v$*, the following inequality holds:*

$$b \geq v$$

Other results for block designs are given by the following theorems from Tonchev [55],

Theorem 6 (Tonchev) *For a 2-*(v,k,λ) *design with* s *identical blocks, the following inequality holds*

$$b \geq sv \tag{2.4}$$

Theorem 7 (Tonchev) *For a 2-*(v,k,λ) *design, the following inequality holds*

$$b \geq \frac{vr^2}{\lambda(v-1)+r} \tag{2.5}$$

Theorem 8 (Tonchev) *The number of blocks in a 2-*(v,k,λ) *design which are not disjoint from a given block* $B \in \mathcal{B}$ *is greater than or equal to*

$$\frac{k(r-1)^2}{(k-1)(\lambda-1)+r-1} \tag{2.6}$$

Wilson [62] strengthened some inequalities concerning the structure of balanced incomplete block designs, like the above, for the case of t-designs with $t \geq 4$.

Packing and Covering designs are other combinatorial designs that can be viewed as a relaxation of t-designs in the following sense. If the t-balanced condition of a t-design, that every t-set of X appears in exactly λ blocks, is modified by replacing the word "exactly" by the word "at most", then we will have the definition of a packing design. In the same way, if "exactly" is replaced by "at least", then we will be defining a covering design. The maximum (respectively minimum) number of blocks used in the packing (respectively covering) design is called the packing (respectively covering) number. The number of blocks b of a t-design corresponds to both a maximum packing number and a minimum covering number.

Schönheim (see [41]) established upper and lower bounds for packing numbers and covering numbers, respectively. His results are the following theorems.

Theorem 9 (Schönheim Upper Bound) *The packing number, $P_\lambda(v,k,t)$, satisfies*

$$P_\lambda(v,k,t) \leq \left\lfloor \frac{v}{k} \left\lfloor \frac{v-1}{k-1} \cdots \left\lfloor \frac{\lambda(v-t+1)}{k-t+1} \right\rfloor \right\rfloor \right\rfloor \tag{2.7}$$

Theorem 10 (Schönheim Lower Bound) *The covering number, $C_\lambda(v,k,t)$, satisfies*

$$C_\lambda(v,k,t) \geq \left\lceil \frac{v}{k} \left\lceil \frac{v-1}{k-1} \cdots \left\lceil \frac{\lambda(v-t+1)}{k-t+1} \right\rceil \right\rceil \right\rceil \tag{2.8}$$

II.1.2. Applications

We will briefly mention a few applications of t-design problems. For surveys in this topic see Beth et al. [8], the CRC Handbook of Combinatorial Designs [13], and Colbourn [16].

II.1.2.1. Codes

Combinatorial designs are highly related to *coding theory* [56, 55, 64]. A *binary code* C of length n is a set of n-tuples with components from {0,1}. Each n-tuple in the code is called *codeword*. The number of codewords in the code is called the *size*. The *distance* between any two codewords is the number of components in which they differ. The distance of any codeword with respect to the zero vector is called its *weight*. An important parameter of a code is its *minimum distance* d, defined as the smallest distance between any two codewords in the code. Suppose that the codewords of C are used as communication messages that are transmitted via a noisy channel, which result in random changes in the values of some components in the transmitted codeword. Let the number of errors (the number of components being changed) be at most e. Then a code with minimum distance d is able to correct $e = (d-1)/2$ errors, that is called *error correcting ability* of the code. The correction is done at the receiver, considering any received word as the image of the codeword form C that is closest to it in terms of distance.

The optimization problems in constant weight codes are finding optimal values for the three parameters, with d and $|C|$ as large as possible, and n as small as possible. Optimal solutions are rarely known in general, but are closely related to t-designs. For example, consider the code C to be the point-block matrix of a 2-design, and let the rows be the codewords. The weight of each codeword is the point replication r of the 2-design. The balanced property of a 2-design, that every two points meet in exactly λ blocks, implies that every two codewords have at least λ common positions. That is, that the distance between any two codewords will be $2(r-\lambda)$. Some specific codes and their related design are: the *Hamming code* to a 2-($\frac{q^m-1}{q-1}, 3, q-1$) design, for any prime power q; the *Ternary Golay code* G_{11} to a 4-(11,5,1); the *Binary Golay code* G_{23} to a 4-(23,7,1). In general, an optimal equidistant q-ary code exists if and only if there exists a resolvable 2-($|C|, |C|/q, n-d$) design [56].

II.1.2.2. Cryptography

A significant application for designs is problems of secure and authenticated message transfer or communication. Optimal solution to these types of problems are very closely related to combinatorial designs (see for example [50, 52]). In several cases, in addition to protect messages against errors occurring in the communication channel, authentication of the sender is desired. We will mention two types of *authentication codes* and their relation to t-designs: general authentication codes and authentication codes with secrecy. The following example we quote here is from Gopalakrishnan and Stinson [26]. Suppose we have two people, Alice and Bob that need to communicate over an insecure channel in such a way that an observer, Oscar, cannot understand what is being said. Oscar can attack the communication by either introducing his own messages, called *impersonation*, or by modifying the existent messages, called *substitution*. The purpose of the authentication code is to protect the integrity of the message, so when Bob receives a message from Alice, he can be sure that it was really sent by Alice and that it was not altered by Oscar. Alice and Bob have a secret *key* chosen in advance. For each key there is an authentication rule, used first by Alice to produce an *authenticator*, and then used by Bob to authenticate the message. Assume Alice and Bob choose a probability distribution for their *authentication strategy*, then a *deception probability by impersonation*, p_1, and a *deception probability by substitution*, p_2, can be computed.

One is interested in minimizing the deception probabilities as well as the number of authentication rules for the code. For general authentication codes there exist a formula for lower bounds on deception probabilities (see [26]). Codes satisfying those bounds with equality and having the minimum number of authentication rules have been characterized in terms of 2-designs with $\lambda = 1$ (Steiner

2-designs). An extension of the impersonation/substitution model is the following. Suppose the encoding rule is used for the transmission of t messages, which Oscar observes in the channel. Then, suppose Oscar inserts a new message of his own, hoping to have it accepted by Bob. This is called *spoofing of order t*. There is a formula for the least number of authentication rules needed to have a code t-fold secure against spoofing. The bound is exact only if the authentication matrix is a $(t+1)$-$(v,k,1)$ design, where k is the number of equiprobable source states, having v possible messages and $\binom{v}{t+1}/\binom{k}{t+1}$ equiprobable authentication rules.

II.1.2.3. Scheduling tournaments

Designs have applications in *scheduling tournaments* like round robin tournaments, oriented, balanced, room squares, arrays, whist, bridge and elimination tournaments. In a *round robin tournament*, every team plays every other team once. If there are $2n$ teams, and the $n(2n-1)$ games are to be arranged into $2n-1$ rounds, with each team playing in one game in each round, we seek a resolvable 2-(2n,2,1) design [4]. The parallel classes correspond to the rounds. That is, the solution will be a schedule for $2n$ teams in $2n-1$ rounds, each round consisting of n games, with each pair of teams meeting exactly once. A variation of this problem is the following: suppose that $2n-1$ pitches of different quality are available. The question is if the games can be allocated to the pitches in such a way that each team will play exactly once on each pitch. This corresponds to finding a 2-(2n-1,n,2) design.

Another tournament arrangement is a *whist tournament* for $4n$ players, denoted Wh($4n$) [4]. It is defined as a schedule of games each involving two players against two others such that: (a) the games are arranged in $4n-1$ rounds, each of n games (b) each player plays in one game per round; (b) each player partners every other player exactly once; (d) each player opposes every other player exactly twice. A solution to a whist tournament corresponds to a resolvable 2-(v,4,1) design. Since such designs exist only if $v \equiv 4 (\text{mod } 12)$, then a Wh($12m + 4$) is a resolvable 2-(12$m$+4,4,1) design. In the case of even number of players $4n + 1$, a Wh($4n+1$) is defined as above, but with the conditions (a) and (b) modified to: (a') the games are arranged in $4n+1$ rounds, each of n games; (b') each player plays in one game in all but one of the rounds. Then a Wh($4n + 1$) corresponds to a nearly resolvable 2-($4n + 1$,4,1) design.

II.1.2.4. Statistical experiments

One of the first applications of combinatorial designs was in statistical design of experiments. Suppose that in certain experiment using treatments and blocks (for example, test of several fertilizers on different plants in agriculture), we may not be able to run all the treatment combinations in each block. Situations like this usually occur because of shortages in experimental apparatus or facilities, or the physical size of the block [42]. For this type of problem it is possible to use 2-designs in which every treatment is not present in every block. The balanced property of the 2-design guarantees that every pair of treatments appears together an equal number of times. By using 2-designs in this type of problems, the total variability may be partitioned into variability form treatments, blocks, and error. Then a statistical model is formulated and solved to estimate the treatment effects. The statistical design of experiments applies to a wide type of experiments, not only agricultural.

II.1.2.5. Network interconnection

Designs are useful for the development of algorithmic and architectural constructions in different areas of computer science. We present an example of multiprocessor interconnection. The interconnection mechanism in multiprocessor systems, using either message passing or shared memory, is based in the same principle: shipping a data item form a node where it is stored to a node where it is needed for processing [7]. Combinatorial designs can be applied to the construction of bus interconnection networks. The t-balanced property, in particular 2-balanced or pairwise balanced for block designs, provides a direct link between any pair of processing elements.

Consider a network with a set of processing elements interconnected by a collection of buses. The representation of this network as a block design is suggested by Berkovich [7]. The points (the elements of the original set) represent the interconnection links, like buses. The blocks (the subsets of these elements) represent the processing nodes. We quote from Berkovich [7] the description of the interconnection mechanism.

> An object stored in the system is associated with a certain link an is replicated in each of the nodes which this link connects. Because of the pairwise balanced design property, for any pair of objects there exists a processing node where copies of these objects reside together. An act of interaction of two objects occurs locally between their copies within this processing node. After the interaction the updates can be immediately sent to all copies of the objects through separate interconnecting links. In the suggested structure, all the copies of interacting objects are update simultaneously.

Without the pairwise balance property of a block design structure, the data will be in different versions and the system will need to utilize a time-consuming procedure to keep track of the time stamps of the copies.

Another example of application of block designs is in the design of congestion-free networks. The problem, as defined by Colbourn [12] is as follows. Connect n network sites using multidrop communication links (for example Ethernet), such that every two sites appear together on at least one link, subject to the constraints that no link has more than k sites on it, and no site appears on more than r links. Optimal solutions to this problem are 2-$(v, k, 1)$ designs (block designs), and 2-$(v, K, 1)$ designs (pairwise balanced designs). The points of the design correspond to the sites, and the blocks correspond to the links. A pairwise balanced design has block sizes from a set K, instead of a fixed size k.

One more example of application of block designs is *quorum systems*. As defined by Colbourn et al. [14], a quorum system is a set system in which any two subsets have nonempty intersection. Quorum systems are used to maintain consistency in distributed systems, like distributed databases. Certain subsets, called *quorums* are identified, then by accessing and updating quorum elements, the intersection property ensures that any quorum contains at least one element that is up-to-date, and the consistency of the system is maintained over time. Important attributes of a quorum system include load, balancing ratio, rank, and availability. Near optimal values for these parameters are obtained in quorum systems constructed from block designs.

II.1.3. Classical construction methods

The classical known methods of construction of designs can be divided into two types: direct and recursive.

Direct methods are often applicable for parameters of a special type and generally use finite groups or fields. For example, permutation groups are used in constructing designs which are invariant under a given automorphism group. Some Steiner systems are constructed from Mathieu groups [55].

The recursive methods developed by Hanani [28] and others, with few exceptions, are applicable mainly to 2-designs (block designs). They yield designs constructed from designs with smaller parameters. This technique is called extension of designs. The asymptotic theory of existence of block designs was established using recursive methods.

Recursive methods combined with appropriate direct constructions are applicable for a wide class of parameters. Using such methods, some necessary conditions were proven sufficient for the existence of certain block designs [55].

II.2. Brief review of polyhedral theory and algorithms

As pointed out by Graham et al. [27], an approach that has been successful in solving some combinatorial optimization problems is to translate them into optimization problems over polyhedra and utilize linear programming techniques. The general linear programming problem is very well-known. Here we include the definition as stated by Korte and Vygen [36]:

LINEAR PROGRAMMING	
Instance:	A matrix $A \in \mathbb{R}^{m\times n}$ and column vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.
Task:	Find a column vector $x \in \mathbb{R}^n$ such that $Ax \leq b$ and $c^T x$ is maximum, decide that $\{x \in \mathbb{R}^n : Ax \leq b\}$ is empty, or decide that for all $\alpha \in \mathbb{R}$ there is an $x \in \mathbb{R}^n$ with $Ax \leq b$ and $c^T x > \alpha$.

A *linear program (LP)* is an instance of the above problem. We often write a linear program as max $\{c^T x : Ax \leq b\}$. A *feasible solution* of an LP max $\{ c^T x : Ax \leq b \}$ is a vector x with $Ax \leq b$. A feasible solution attaining the maximum is called an *optimal solution*. The set of all feasible solutions $P = \{ x \in \mathbb{R}^n : Ax \leq b \}$ of an LP is the intersection of finitely many halfspaces and is called a *polyhedron*. The following definition is from Korte and Vygen [36].

Definition A *polyhedron* in $\mathbb{R}^n$ is a set of type $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ for some matrix $A \in \mathbb{R}^{m \times n}$ and some vector $b \in \mathbb{R}^m$. If A and b are rational, then P is a *rational* polyhedron.

A polyhedron $P \in \mathbb{R}^n$ is said to be *bounded*, if P is a bounded subset of $\mathbb{R}^n$, i.e. if there exists $r \in \mathbb{R}$ such that $\|x\| \leq r$ for every $x \in P$ [6]. A bounded polyhedron is also called a *polytope*. The following theorem supports the essence of polyhedral combinatorics, for a proof see Cook et al. [17].

Theorem 11 (see [17]) *A set P is a polytope if and only if there exists a finite set V such that P is the convex hull of V.*

A general method for obtaining min-max relationships to prove optimality of solutions to combinatorial optimization problems is given by Cook et al. [17] and quoted here:

- Represent the combinatorial problem as an optimization problem over a finite set of vectors S, for example, by considering characteristic vectors.
- Find a linear description of the convex hull of S.
- Apply the duality theory of Linear Programming to obtain a min-max relation for the combinatorial problem.

Most research in polyhedral methods focuses in establishing methods to accomplish the second step, which is usually by far the most difficult of the three. As pointed out by Graham, Grötschel and Lovász [27], finding a description of the convex hull by linear equations and inequalities is by no means a simple task. By problem specific investigations one can often find classes of valid and even facet-defining inequalities that partially describe the polyhedra of interest.

Although some combinatorial optimization problems can be formulated as LPs, most of them require the condition on the variables to be integers. Linear programs with integrality constraints are called *integer programs (IP)*.

The set of feasible solutions of an IP can be written as $\{x : Ax \leq b, x \in \mathbb{Z}^n\}$ for some matrix A and some vector b. Let the polyhedron $\{x : Ax \leq b\}$ be P, and define $P_I = \{x : Ax \leq b\}_I$ as the convex hull of the integral vectors in P. We call P_I the *integer hull* of P, and $P_I \subseteq P$.

II.2.1. Integrality of polyhedra

There are certain conditions under which polyhedra are integral (that is, $P = P_I$) and in this case the IP is equivalent to the LP relaxation, and can hence be solved in polynomial time.

Definition A matrix A is *totally unimodular (TU)* if each subdeterminant of A is 0, $+1$, or -1.

In particular, each entry of a totally unimodular matrix must be 0, $+1$, or -1. One important result in integer programming is the following,

Theorem 12 (Hoffman and Kruskal, 1956) *An integral matrix A is totally unimodular if and only if the polyhedron $\{x : Ax \leq b, x \geq 0\}$ is integral for each integral vector b.*

In other words, when the constraint matrix A of a problem is TU, the linear programming relaxation solves the Integer Programming problem. Some examples of IP problems that have this property are shortest path problem, maximum flow problem, minimum spanning tree problem, matching problem, assignment problem.

In general, however, Integer Programming is much harder than Linear Programming, and polynomial-time algorithms are not known. As mentioned by Korte and Vygen [36], this is indeed not surprising since virtually all apparently hard problems can be formulated as integer programs.

II.2.2. Complexity and problem reductions

In Section II.2.1, we mentioned some integer programming problems that have nice properties that make then solvable in polynomial time, say by linear programming, or other efficient algorithms. These are therefore considered "easy" problems. However, no efficient polynomial time algorithm has been found for other problems like 3-satisfiability, 3-dimensional matching, vertex cover, clique, hamiltonian circuit, partition, among many others, and therefore are considered "difficult". A well-known theory that formalizes a classification on how computationally easy or difficult is to decide if a problem has a solution is called the *Theory of $\mathcal{NP}$-completeness*. This was laid originally in a 1971 paper of Stephen Cook, and widely spread through the book of Garey and Johnson [25]. An updated book in this topic is of Ausiello, et al. [5]. We quote from [25]:

> The theory of $\mathcal{NP}$-completeness provides many straightforward techniques for proving that a given problem is "just as hard" as a large number of other problems that are widely recognized as being difficult and that have been confounding the experts for years.

Most complexity theory is based on decision problems. These can be regarded as those problems having *yes-no* answers. An optimization problem of the form max$\{cx : x \in S\}$ can be posed as a decision problem: is there an $x \in S$ such that $cx \geq k$ for a given integer k?

The class $\mathcal{NP}$ is the class of decision problems with the property that for any instance for which the answer is *yes*, there is a polynomial proof of the *yes*. The name $\mathcal{NP}$ stands for *nondeterministic polynomial*. The class $\mathcal{P}$ is the class of decision problems in $\mathcal{NP}$ for which there exists a polynomial time algorithm. So $\mathcal{P}$ is the class of "easy" problems. It is not known if $\mathcal{P} = \mathcal{NP}$.

The theory of $\mathcal{NP}$-completeness is based on a special kind of polynomial time reduction: Let P and Q be decision problems. It is said that P *polynomially transforms* to Q if there is a function from the set of instances of P to the set of instances of Q computable in polynomial time such that yes-instances of P are transformed to yes-instances of Q, and no-instances of P are transformed to no-instances of Q. Polynomial transformations are sometimes called *Karp reductions* [36]. This reduction is useful in the sense that if P polynomially reduces to Q and there is an efficient algorithm for Q, then there is an efficient algorithm for P.

Definition A decision problem $P \in \mathcal{NP}$ is called $\mathcal{NP}$*-complete* if all other problems in $\mathcal{NP}$ polynomially transforms to P.

The above implies that if there is a polynomial-time algorithm for any $\mathcal{NP}$-complete problem, then the open question would be answered, $\mathcal{P} = \mathcal{NP}$, and all the problems in the class $\mathcal{NP}$ would be solved in polynomial time. Until now, nobody has succeeded in proving that $\mathcal{P} = \mathcal{NP}$ or in showing that $\mathcal{P} \neq \mathcal{NP}$. Given the enormous list of $\mathcal{NP}$-complete problems (for a compendium see [25, 5]), it is believed that $\mathcal{P} \neq \mathcal{NP}$.

II.2.3. Separation and optimization

It has been proved (see [36] for example) that, under some reasonable assumptions, we can optimize a linear function over a polyhedron P in polynomial time (regardless of the number of constraints) if we have the so-called *separation oracle*. This separation oracle is a subroutine for the problem of given a polytope P and a rational vector $y \in \mathbb{Q}^n$, either decide that $y \in P$ or find a rational vector $d \in \mathbb{Q}^n$ such that $dx < dy$ for all $x \in P$. In other words, in theory it does not matter that the size of the linear system (polyhedral description) is by far greater than the original combinatorial optimization problem, the separation oracle makes possible to solve the problem in polynomial time.

The bottom line result is that efficient optimization and efficient separation are equivalent. That is, if the separation problem can be solved in polynomial time, then the optimization problem can be solved in polynomial time. The converse is also true. This equivalence is the guide for searching for more practical and polynomial-time algorithms [17].

II.2.4. Branch-and-cut

A technique widely used to cope with integer programming problems is *cutting planes*, which is a general method for finding the integer hull by successively cutting off parts of $P \backslash P_I$. Although cutting planes does not yield a polynomial-time algorithm, it has proven successful in some cases. It is often combined with an implicit enumeration technique and called *branch-and-cut*. Here we include a description of this technique as found in Wolsey [63].

A *branch-and-cut* algorithm is a branch-and-bound algorithm in which cutting planes are generated through the search tree. The goal is to get a tight dual bound at a node by adding as many cuts as possible to improve the formulation. In practice, there is a trade-off. If many cuts are added at each node, re-optimization may be much slower, and keeping all the information in the tree would be more difficult. Unlike branch-and-bound where the problem to be solved at each node is obtained by just adding bounds, in branch-and-cut it is necessary to remember both the bounds and the added cuts at each node. A flowchart of the basic steps of a branch-and-cut algorithm is shown in Figure 1.

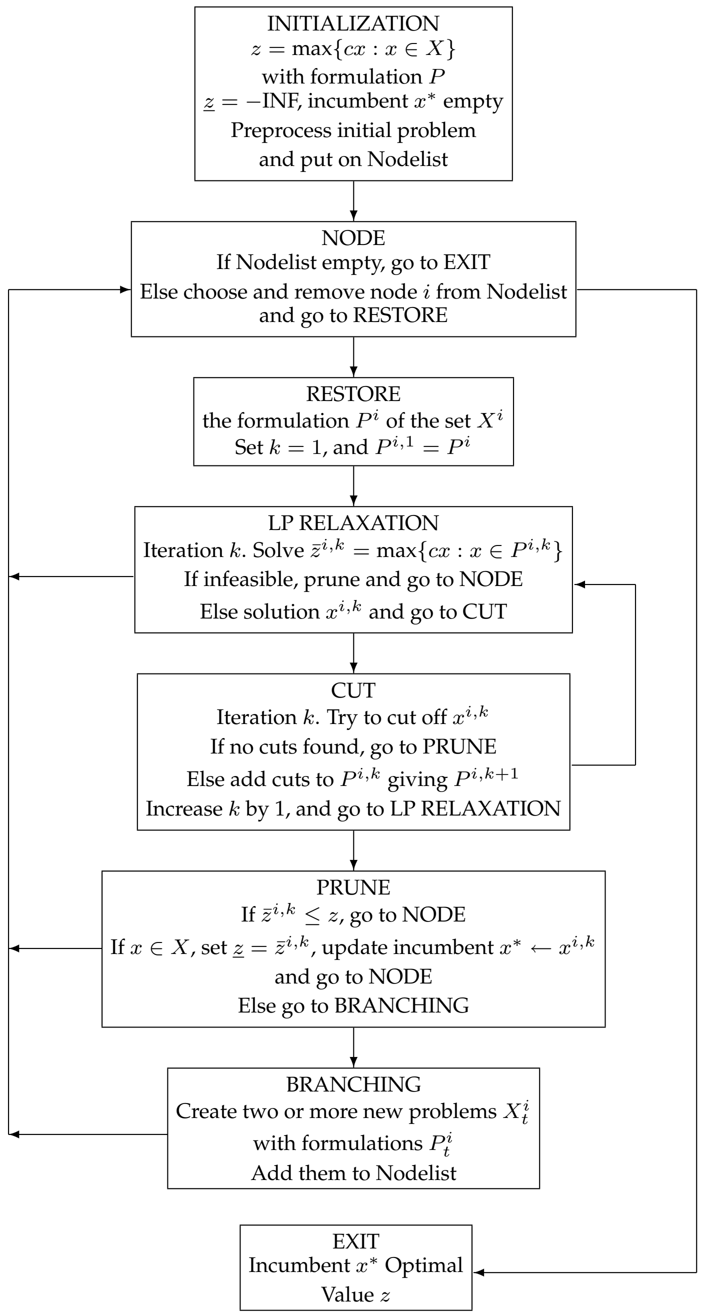

Fig. 1. Branch-and-cut. From *Integer Programming*, L. A. Wolsey, Copyright ©1998 by John Wiley & Sons. Reprinted by permission of John Wiley & Sons, Inc.

CHAPTER III

POLYHEDRAL APPROACH TO COMBINATORIAL DESIGNS: CURRENT RESEARCH STATUS

In this chapter, we summarize relevant work found in the literature related to computational design theory, with special emphasis on polyhedral studies of combinatorial designs. We found basically two existent integer programming formulations for the problem of finding a t-design, which we describe in Section III.1. Studies have been done on both of them, some using special cases of the parameters (like for a specific t value, or λ value) or variations on the formulation (like relaxations on the variables or modifications on the constraints). We describe these efforts in slightly more detail in Section III.2. In Section III.3, we include some of the current open problems in t-designs and related combinatorial design problems. The amount of open problems clearly highlights the richness of the research area. Finally, although we do not deal with exhaustive search or randomized search methods in our work, we also describe in Section III.4 some other recent computer search work applied to t-designs. It is interesting to look at these examples to have an idea of the magnitude of the task of computationally finding combinatorial designs, and to realize why there are many open problems.

III.1. Existent integer programming formulations

We will describe existent integer programming formulations for the problem of finding a combinatorial design t-(v,k,λ). The labels used for the formulations in this section are due to Moura [47]. The incidence structure associated with a t-

design can be represented by a matrix. In the literature, two types of matrix representations are used the most. One is called the *tk-incidence matrix* and the other is called the *point-block incidence matrix*. They are defined as follows.

Definition The *tk-incidence matrix*, A, associated with a t-(v, k, λ) design, is a (0-1) matrix of $\binom{v}{t}$ rows and $\binom{v}{k}$ columns. The rows of A are indexed by the set I, which corresponds to all the t-subsets of points. The columns of A are indexed by the set J, which corresponds to all the k-subsets of points (i.e. the blocks). The elements of A are $a_{ij} = 1$ if the t-set i is included in the k-set j; 0 otherwise.

Definition The *point-block incidence matrix*, D, associated with a t-(v, k, λ) design with b blocks is a (0-1) matrix of v rows and b columns. The elements of D are $d_{ij} = 1$ if the point i is included in the block j; 0 otherwise.

A natural integer programming formulation for t-designs that uses the tk-incidence matrix, A, is a *set-partitioning* type of formulation.

$$\begin{aligned} Ax &= \lambda, \\ x &\in \{0, 1\}^{\binom{v}{k}} \end{aligned} \tag{DP}$$

The constraints of the linear system on binary variables (DP) ensure that every t-subset of points is contained in exactly λ blocks. If the solution set of the system (DP) is nonempty, then a t-design exists. A solution vector x of the above system represents the *incident vector* associated with the t-design, which is defined by:

$$x \in \mathbb{B}^{\binom{v}{k}} : x_j = \begin{cases} 1, \text{ if } j \text{ is a block in the design} \\ 0, \text{ otherwise} \end{cases} \tag{3.1}$$

A set-partitioning model has two related optimization models: the *set-packing* and the *set-covering* models. The optimal maximum packing corresponds to the optimal minimum covering, and both corresponds to the solution of the partitioning model (DP). Therefore the problem of finding a t-design may be formulated as either a set-packing or a set-covering optimization problems as follows.

$$\begin{aligned} \text{maximize} \quad & 1^T x \\ \text{subject to} \quad & Ax \leq \lambda \\ & x \in \{0,1\}^{\binom{v}{k}} \end{aligned} \tag{PDP}$$

$$\begin{aligned} \text{minimize} \quad & 1^T x \\ \text{subject to} \quad & Ax \geq \lambda \\ & x \in \{0,1\}^{\binom{v}{k}} \end{aligned} \tag{CDP}$$

The following proposition by Moura [45] relates the problems described above.

Proposition 13 (Moura) *Assume that a t-(v, k, λ) design exists. Let $x^* \in \mathbb{R}^{\binom{v}{k}}$. Then the following statements are equivalent:*

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

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

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

Another integer programming formulation for a t-(v, k, λ) design tries to construct the point-block incidence matrix D. It has some non-linear inequalities. In this case, the incidence vector associated with the t-design corresponds to the elements of the matrix D. We will denote the incidence vector as y. As previously defined, the total number of blocks is equal b, and the number of blocks containing a given point is equal to r, then y is defined as:

$$y \in \mathbb{B}^{vb} : y_{ij} = \begin{cases} 1, \text{ if point } i \text{ is in block } j \text{ of the design} \\ 0, \text{ otherwise} \end{cases} \tag{3.2}$$

The formulation for the special case of 2-designs has some quadratic constraints and is defined as follows,

(QDP)

$$\sum_{j=1}^{b} y_{ij} = r, \qquad 1 \le i \le v \tag{qa}$$

$$\sum_{i=1}^{v} y_{ij} = k, \qquad 1 \le j \le b \tag{qb}$$

$$\sum_{l=1}^{b} y_{il} y_{jl} = \lambda, \qquad 1 \le i < j \le v \tag{qc}$$

$$y \in \{0, 1\}^{vb}$$

Equations (qa) are the *point degree* constraints which ensure that every point appears exactly in r blocks. Equations (qb) are the *block size* constraints which ensure that every block has exactly k points. Either constraints (qa) or constraints (qb) can be eliminated from the system, since integer solutions of the remaining system will satisfy the eliminated also [47]. This is a consequence of the definitions of r and b [8]. In [47], Moura indicates that the quadratic formulation (QDP) can be generalized for t-designs with $t > 2$ by having a system of equations of degree t instead of the quadratic system of equations (qc).

III.2. Recent polyhedral applications

Only relatively recent studies of polyhedral methods applied to combinatorial designs have been pursued. We will mention, to the best of our knowledge, the only polyhedral studies both theoretical and computational applied to combinatorial designs found in the literature. The earliest we have reference of is a 1986 doctoral thesis by Zehendner (see [48]) on strong upper bounds for block codes.

Moura [47] studied polyhedral aspects of combinatorial designs in a 1999 doctoral dissertation, and also published those results in [46]. She studied the set-packing problem (PDP) applied to t-designs with $\lambda = 1$. The set-packing polytope in general has been widely studied in the literature, and classes of valid inequalities have been derived for it. Based on the fact that the solution of a maximum optimal packing will be a t-design, and the fact that nearly-optimal solutions will be packing designs, Moura implemented a branch-and-cut algorithm and replicated results for some instances of packings and t-$(v, k, 1)$ designs. The test instances used are shown in Table 2.

According to Moura [47], Wengrzik in a 1995 master's thesis also studied a cutting-plane approach to 2-designs (block designs) and was able to replicate solutions for some instances. The starting model used was (QDP), but with the

Table 2. Test instances used by Moura [46] in a branch-and-cut algorithm

t	v	k	λ	b	Type
2	5	3	1	2	Packing
2	6	3	1	4	Packing
2	7	3	1	7	Design
2	8	3	1	8	Packing
2	9	3	1	12	Design
2	10	3	1	13	Packing
2	11	3	1	17	Packing
2	12	3	1	20	Packing
2	13	3	1	26	Design
2	14	3	1	27	Packing
3	8	4	1	14	Design
3	10	4	1	30	Design
3	14	4	1	91	Design

quadratic constraints linearized by introducing new binary variables for each possible combination of pairs of points in every block. According to Moura [47], Wengrzik's model based on (QDP) uses additional binary variables of the form $z_{ijl} = y_{il}x_{jl}$ as shown in equation (LDP).

Margot [39] studied covering design problems using the set-covering model (CDP) and proposed a branch-and-cut approach on it for covering designs with $\lambda = 1$. He used an original isomorphism pruning of the enumeration tree. The test instances that solved efficiently with his approach were all covering design problems with $v \leq 10$. He also tested four open instances of covering designs, one of them successfully solved, and the others with $v = 11$ remaining unsolved. Table 3 shows the parameters of the open instances tested by Margot.

$$\begin{aligned} \sum_{i=1}^{v} y_{ij} &= k, && 1 \leq j \leq b \\ \sum_{l=1}^{b} z_{ijl} &= \lambda, && 1 \leq i < j \leq v \\ z_{ijl} - y_{il} &\leq 0, && 1 \leq i < j \leq v, 1 \leq l \leq b \\ z_{ijl} - y_{jl} &\leq 0, && 1 \leq i < j \leq v, 1 \leq l \leq b \\ y_{il} + y_{jl} - z_{ijl} &\leq 1, && 1 \leq i < j \leq v, 1 \leq l \leq b \\ y_{ij}, z_{ijl} &\in \{0,1\}, && 1 \leq i \leq v, 1 \leq j \leq b, 1 \leq l \leq b \end{aligned} \tag{LDP}$$

Table 3. Open test instances used by Margot [39] in a branch-and-cut algorithm

t	v	k	λ	b	Type	Result
4	10	5	1	≤ 51	Covering	Solved, b=51
4	11	6	1	≤ 41	Covering	Open
5	11	6	1	≤ 100	Covering	Open
5	11	7	1	≤ 34	Covering	Open

Mannino and Sassano [38] proposed a branch-and-bound algorithm to solve hard instances of set covering problems arising from 2-(v,3,1) designs (also called Steiner triple systems STS_v, see Table 1). The incidence matrix they used is similar to the transpose of a point-block incidence matrix. For a STS_v, the incidence matrix A_v has (i) v columns; (ii) every row has exactly $k = 3$ ones; (iii) for every pair of columns j and l, there is exactly one row i such that $a_{ij} = a_{il} = 1$. These conditions on A_v imply that the number of rows corresponds to the number of blocks b of the 2-design. That is, in this case $b = v(v-1)/6$. The optimal minimum (best possible) would be the corresponding value of r for the 2-design, in this case of STS_v would be $r = (v-1)/2$. The best covering number obtained will be, of course, greater or equal than r. The test instances used are listed in Table 4. With their algorithm, they were able to replicate solutions to STS_{27}, STS_{45}, and STS_{81}; and to improve both the covering number of STS_{135} from 105 to 104, and the covering number of STS_{243} from 203 to 202.

Table 4. Test instances used by Mannino and Sassano [38] in a branch-and-cut algorithm

t	v	k	λ	Type	Best Number	Name
2	27	3	1	Covering	18	STS_{27}
2	45	3	1	Covering	30	STS_{45}
2	81	3	1	Covering	60	STS_{81}
2	135	3	1	Covering	104	STS_{135}
2	243	3	1	Covering	202	STS_{243}

It is worth to note that STS_{27} and STS_{45} correspond, respectively, to problems *Stein27* and *Stein45* included in the problem set of MIPLIB 3.0 [9]. The only difference is that the constraint matrix of *Stein27* or *Stein45* has an extra row: a coupling constraint that sums all the variables and enforces the sum value to be greater than or equal to r. The library MIPLIB is a standard test set of rather difficult instances used to compare the performance of mixed integer optimizers.

III.3. Some open problems

The basic problem in design theory is to find sufficient conditions for the existence of t-designs [8], as well as methods for constructing them [59]. There are still many open problems in combinatorial designs. In the particular case of block designs, an asymptotic existence theory due to Wilson (see [8, 61]) establishes that the necessary conditions of Corollary 4 are sufficient for the existence of block designs for v sufficiently large with respect to k. Despite the fact that existence conditions for block designs are well understood in an asymptotic sense, complete solutions are only known for block designs with k=3, 4 and 5 [8].

For the case $k = 5$ and $\lambda = 1$, the existence of *resolved* block designs for some values of v is still undecided. In 1996 [1] the list was published with 43 open cases, but was reduced to 5 open cases in 1997 [3]. Table 5 shows the values of v for which the existence of 2-(v,5,1) RBIBD remains open.

Table 5. Values of v for which the existence of a resolved block design 2-(v,5,1) remains open

45	225	345	465	645

Also, for the case $k = 5$ and $\lambda = 4$ there are some values of v for which the existence of resolved block designs is not know. The list of 10 open cases published in 1996 [1] was reduced to 9 in 1997 [21] and later to 8 in 2001 by a non-existence proof for the smallest open case $v = 15$ by [34]. Table 6 shows the values for which the existence is still unknown.

For the case $k = 6$ and $\lambda = 1$, the largest open existence case for $v = 2031$ was settled affirmatively in 1997 [21] jointly with some other open values of v, making

Table 6. Values of v for which the existence of a resolved block design 2-(v,5,4) remains open

70	90	95	135	160	185	190	195

the largest open case today the value of $v = 801$. The smallest open case of $v = 46$ was solved in 2001 by [31] with a result of non-existence. Table 7 shows the values of v for which the existence is still undecided [2].

Table 7. Values of v for which the existence of a block design 2-(v,6,1) remains open

51	61	81	166	226	231	256	261
286	291	316	321	346	351	376	406
411	436	441	471	496	501	526	561
591	616	646	651	676	771	796	801

For other small values of k, Table 8 shows some parameter sets for which the existence of a block design is still open. The sample is limited to values of $v \leq 100$ and $r \leq 41$ from [40] considering the update given online by Dinitz [21]. The table is sorted by increasing v.

There are also open problems for t-designs with $\lambda = 1$, called Steiner systems, see [15]. As pointed out by Dinitz and Stinson [22], the construction of a 6-$(v, k, 1)$ design remains one of the outstanding open problems in the study of t-designs.

In addition to the open problems in t-designs, there are also open problems in other combinatorial designs including, but not limited to, packing and covering designs. Recall that an optimal packing or an optimal covering is a t-design for admissible parameters, but solutions to packings and coverings can also be considered relaxations or approximations to t-designs. According to Stinson [54],

Table 8. Some open problems in block designs for small k, $v \leq 100$ and $r \leq 41$

No. in [40]	v	b	r	k	λ
78	22	33	12	8	4
225	39	57	19	13	6
88	40	52	13	10	3
297	40	60	21	14	7
1180	42	123	41	14	13
329	45	66	22	15	7
261	45	75	20	12	5
133	46	69	15	10	3
841	46	161	35	10	7
943	49	84	36	21	15
48	51	85	10	6	1
426	51	85	25	15	7
859	51	85	35	21	14
937	52	104	36	18	12
945	55	90	36	22	14
209	55	99	18	10	3
501	55	99	27	15	7
383	55	132	24	10	4
495	55	135	27	11	5
916	55	198	36	10	6
760	56	88	33	21	12
263	57	76	20	15	5
556	57	76	28	21	10
552	57	84	28	19	9
1130	57	152	40	15	10
646	58	87	30	20	10
1020	58	116	38	19	12
73	61	122	12	6	1

Table 8. Continued

No. in [40]	v	b	r	k	λ
669	63	93	31	21	10
863	64	80	35	28	15
765	64	96	33	22	11
164	65	80	16	13	3
1146	65	104	40	25	15
708	65	160	32	13	6
1064	66	99	39	26	15
848	66	165	35	14	7
396	69	92	24	18	6
804	69	102	34	23	11
936	69	138	36	18	9
947	70	105	36	24	12
344	70	161	23	10	3
982	75	111	37	25	12
1063	76	114	39	26	13
420	76	190	25	10	3
1019	77	154	38	19	9
1008	77	266	38	11	5
764	78	117	33	22	9
542	78	182	28	12	4
169	81	81	16	16	3
716	81	162	32	16	6
152	81	216	16	6	1
568	85	85	28	28	9
212	85	102	18	15	3
860	85	119	35	25	10
394	85	136	24	15	4
105	85	170	14	7	1

Table 8. Continued

No. in [40]	v	b	r	k	λ
637	85	170	30	15	5
301	85	105	21	17	4
928	85	204	36	15	6
540	85	238	28	10	3
580	88	116	29	22	7
388	89	178	24	12	3
505	91	117	27	21	6
948	91	126	36	26	10
1150	91	130	40	28	12
942	91	156	36	21	8
631	91	210	30	13	4
296	92	138	21	14	3
547	92	184	28	14	4
847	92	230	35	14	5
723	93	124	32	24	8
1145	93	155	40	24	10
226	96	114	19	16	3
645	96	144	30	20	6
856	96	168	35	20	7
1011	96	228	38	16	6
779	97	97	33	33	11
715	97	194	32	16	5
557	99	126	28	22	6
851	99	231	35	15	5
496	100	225	27	12	3
210	100	150	18	12	2

essentially nothing is known about the packing numbers for parameters $t = 3$, $k = 4$, $\lambda = 1$, all $v \equiv 5 (\text{mod } 6)$, $v > 17$. Also Stinson [53], lists some unknown covering number problems. Recent results on upper bounds for covering numbers are the new bounds derived by Bluskov and Heinrich [10]. Tables 9 and 10 include a few open problems with small parameters (see [41, 46, 53, 54]).

Table 9. Some unknown packing number problems for small parameters

t	v	k	λ
3	18	5	1
3	19	5	1
3	20	5	1
3	21	5	1
4	12	5	1
4	13	5	1
4	14	5	1
4	17	6	1
4	18	6	1
4	18	7	1
4	19	7	1
5	13	6	1
5	14	6	1
5	15	6	1
5	15	7	1
5	16	7	1

Table 10. Some unknown covering number problems for small parameters

t	v	k	λ
2	28	5	1
2	19	6	1
3	12	5	1
3	13	5	1
3	13	6	1
3	14	6	1
3	15	5	1
3	15	6	1
3	16	5	1
3	16	6	1
3	16	7	1
3	17	6	1
3	17	7	1
3	18	7	1
3	19	4	1
3	19	5	1
3	19	6	1
3	19	7	1
3	20	5	1
3	20	6	1
3	20	7	1
4	12	6	1
4	12	7	1
4	13	5	1
5	11	6	1
5	11	7	1
5	12	7	1

III.4. Other computational approaches

One way to settle these existence questions is to use computers to either find a solution or to exhaust the search showing that the solution does not exist. However, as pointed out by Furino [24], searching for a new combinatorial design can be an intimidating task. In this section we give some examples of computer search for t-designs.

A noteworthy example is the search for 2-(22,8,4), the 2-design with smallest point set for which it is not known if a solution exists. This problem was the smallest undecided case in some statistical tables published by Fisher and Yates in 1963 for design of statistic experiments. It is still undecided despite the fact that much effort and computation has been devoted to it, including a search that took more than 2 years of CPU time [57].

Another example is a computer search for 2-(46,6,1) by Houghten et al. used to prove that the design does not exist. The search ended after more than 170,000 hours normalized to an execution on a SPARC-10 computer [31]. Other non-existence result for 2-(15,5,4) resolved design was proved by an exhaustive computer search by Kaski and Östergard [34].

Morales [43], using tabu search, constructed six new difference families which lead to six new balanced incomplete block designs: 2-(49,9,3), 2-(61,10,3), 2-(46,10,4), 2-(45,11,5), 2-(85,8,2) and 2-(34,12,10). Using a backtracking algorithm, Morales and Velarde [44] found and listed the five nonisomorphic 2-(12,4,3) resolved balance incomplete block designs.

CHAPTER IV

NEW PROBLEM EQUIVALENCE RESULT

In this chapter we present one of the main contributions of our research work. We prove that the problem of finding a t-design has an equivalent graph problem. This is presented in Section IV.1. This equivalence result leads to another contribution of our work: a novel integer programming formulation for the problem of finding a combinatorial t-design, which we include in Section IV.2. This new formulation is not only an application of an existent IP model (like the set-covering or set-packing models) adapted to be used to find a combinatorial design, but it is a result of a problem equivalence. We also give a polyhedral analysis of the new formulation in Section IV.3, where we conclude, among other results, the dimension of this new polytope for t-designs.

IV.1. Restricted b-factors in bipartite graphs

We begin with a few definitions from graph theory (see [20]). A undirected *graph* G denoted $G = (V, E)$ consist of disjoint finite sets $V(G)$ of *vertices*, and $E(G)$ of *edges*, and a relation associating each edge with a pair of vertices called its *ends*.

Definition For a graph $G = (V, E)$ and $A \subseteq V$, a *graph cut* $\delta(A)$ of G is a set defined as $\delta(A) = \{e \in E : e$ has an end in A and an end in $V \backslash A\}$. For a single vertex $v \in V$, the graph cut, $\delta(v)$, is also called *degree* of the vertex v.

Definition For a graph $G = (V, E)$ and $A \subseteq V$, denote by $\gamma(A)$ the set $\gamma(A) =$

$\{e \in E : \text{ both ends of } e \text{ are in } A\}$.

Definition A graph $G = (V, E)$ is called *complete* if all the vertices of G are pairwise adjacent.

Definition For a graph $G = (V, E)$ and $U \subseteq V$, the *induced subgraph*, denoted $G[U]$, is the graph on U whose edges are precisely the edges of G with both ends in U.

Definition Let $G = (V, E)$ be a (non-empty) graph. The set of *neighbors* of a vertex v in G is denoted by $N_G(v)$, or briefly by $N(v)$. More generally for $U \subseteq V$, the neighbors in $V \setminus U$ of vertices in U are called *neighbors of* U; the set is denoted by $N(U)$.

The general matching problem, *integer b-matching*, in a graph $G = (V, E)$ involve choosing a subset of edges, subject to degree constraints on the vertices and allowing each edge to be chosen a nonnegative integer number of times. The special case where each edge is chosen no more than once (0-1 b-matching) and each of the degree constraints holds at equality (*perfect* 0-1 b-matching) is also called *b-factor*.

Definition Given a graph $G = (V, E)$ and numbers $b : V(G) \to \mathbb{N}$, a *b-factor* is a subset of edges $M \subseteq E$ with the property that each vertex v in the subgraph $G(M) = (V, M)$ is met by exactly b_v edges.

We focus our attention on bipartite graphs. A graph $G = (V, E)$ is called *bipartite* is V admits a partition in two classes such that every edge has its ends in different classes.

Definition A *biclique* is complete bipartite graph, denoted $K_{p,q}$, where p is the number of vertices in one partition and q the number of vertices in the other partition. Trivial biclique graphs of the form $K_{p,1}$ or $K_{1,q}$ are called *stars*.

A *restricted* b-factor is a b-factor that does not have certain forbidden graphs as subgraphs. For example, *k-restricted 2-factor* consists of finding, in a complete graph K_n, a 2-factor with no cycles of length k or less. The case k=3, the *triangle-free 2-factor* problem, was studied by Cornuéjols and Pulleyblank [18]. They mention that the problem if maximizing a linear function over the set of triangle-free 2-factors is of interest because of its relation to the travelling salesman problem. Cunningham and Wang [19] pointed out that the k-restricted 2-factor problem in K_n is equivalent to the symmetric travelling salesman problem when $n/2 \leq k \leq n-1$.

The case k=4, the *square-free 2-factor* was studied by Hartvigsen [29] for bipartite graphs. Those results were later sharpened by Király [35]. Since a cycle of length four in a bipartite graph is a biclique $K_{2,2}$, Frank [23] later derived a formula for the maximum number of edges in a $K_{t,t}$-free t-matching in bipartite graphs. The $\mathcal{NP}$-completeness of some restricted matching problems on bipartite graphs was proved by Itai et al. [33].

For general matching problems, and in particular for the b-factor problem, there exist linear-programming descriptions as well as results on polynomial time solvability [17]. However, the additional constraints of the restricted b-factor problem, as its relation to the travelling salesman problem suggests, make the problem much harder to solve. In fact, Papadimitriou (see [18]) proved that deciding if a graph has a 2-factor with no cycle of length k=5 or less is $\mathcal{NP}$-complete.

In our work, given natural numbers r and k, we consider only b-factors in a complete bipartite graph $G = (L \cup R, E)$ with $b : V(G) \rightarrow \{r, k\}$ defined as,

$$b(v) = \begin{cases} r, & \text{if } v \in L \\ k, & \text{if } v \in R \end{cases} \tag{4.1}$$

We call this a $\{r, k\}$-factor in a bipartite graph. The forbidden subgraphs will be some bicliques. We study the relation of this *biclique-free b-factor* problem to t-designs. We focus on admissible parameters of t-designs, and the result applies to both simple t-designs and designs with repeated blocks. Our equivalence result is the following,

Theorem 14 *The problem of finding a t-(v,k,λ) design with b blocks and point replication r, is equivalent to the problem of finding a $K_{t,\lambda+1}$-free $\{r,k\}$-factor in a complete bipartite graph $K_{v,b}$.*

Proof.

Let $\mathcal{T}$ be the set of t-designs with given parameters (v,k,λ,b,r) and let $\mathcal{F}$ be the set of $K_{t,\lambda+1}$-free (r,k)-factors on a complete bipartite graph $K_{v,b}$. We need to show $\mathcal{T} = \mathcal{F}$. First show $\mathcal{T} \subseteq \mathcal{F}$, which is straightforward. For any $T \in \mathcal{T}$, a t-design by definition is an incidence structure $D = (X, \mathcal{B}, I)$. Construct a bipartite graph $G = (L \cup R, E)$ as follows: create one vertex in L corresponding to every element in X, and one vertex in R corresponding to every element in $\mathcal{B}$. Create and edge $e \in E(G)$ if $(x, B) \in I$ for any $x \in X$, $B \in \mathcal{B}$ (see Figures 2 and 3 for a particular

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1
\end{bmatrix}$$

Fig. 2. A t-design 2-(9,3,1) represented by point-block incidence matrix.

example).

The definition of a t-design implies that the degree of every vertex $v \in L$ is equal to r, and that the degree of every vertex $v \in R$ is equal to k. Then, it corresponds to a $\{r, k\}$-factor on the bipartite graph G. Call this factor M. Also by definition of a t-design, every t-subset of points appears in exactly λ blocks. Taking a subset of edges $A \subset M$ corresponding to the induced subgraph of any t different points and any $\lambda + 1$ different blocks from the t-design, the t-balanced property of a t-design implies that $|A| \leq t(\lambda + 1) - 1$, therefore the factor M is also biclique $K_{t,\lambda+1}$-free.

Now show $\mathcal{F} \subseteq \mathcal{T}$. For any $F \in \mathcal{F}$, the restriction to be a graph $\{r, k\}$-regular makes it satisfy the point replication and block size conditions of a t-design. We call the subset of vertices in $V(F)$ that have degree r, points, and those that have degree k, blocks. Let L be the set of points and R be the set of blocks in the factor

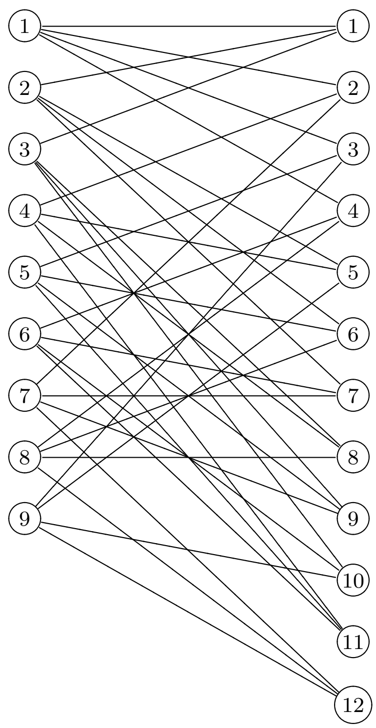

Fig. 3. A biclique $K_{2,2}$-free $\{4, 3\}$-factor in a bipartite graph $K_{9,12}$.

F. The restriction for F to be $K_{t,\lambda+1}$-free clearly prohibits any t-set of points to be incident to more than λ blocks. To show any t-set of points is not incident to fewer than λ blocks, suppose by way of contradiction, that there exists a $F \in \mathcal{F}$ with a t-set of points incident to fewer than λ blocks. Without loss of generality, by Corollary 2 we can consider only the case $s = 2$. That is, by way of contradiction (see Figure 4 for an illustration) we suppose there exists a $S = \{p, q\} \subset L$ with neighbors $N(S) = N(p) \cup N(q) \subseteq R$ such that,

$$|N(p) \cap N(q)| < \lambda_2$$

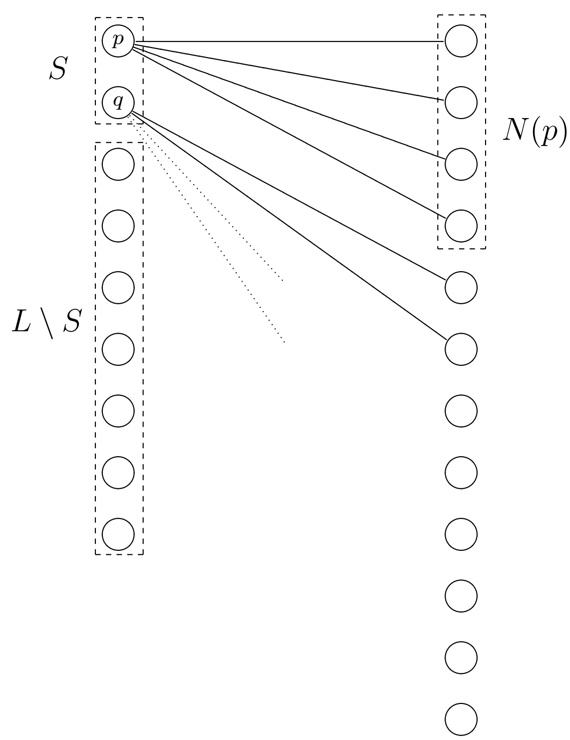

Fig. 4. Illustration for second part of the equivalence proof.

More specific,

$$|N(p) \cap N(q)| = \lambda_2 - 1$$

Then the number of edges with one end in S and the other end in $N(p)$ is,

$$|\gamma(S \cup N(p))| = r + (\lambda_2 - 1) \tag{4.2}$$

We count in two ways the edges having one end in $N(p)$ and the other end in $L \setminus S$. Since the degree of each vertex $v \in R$ is k, the number of edges going from $N(p)$ to the complement of S can be computed using (4.2) as follows

$$|\gamma\big(N(p) \cup (L \setminus S)\big)| = rk - (r + (\lambda_2 - 1)) = r(k-1) - (\lambda_2 - 1) \tag{4.3}$$

Now, we count the number of edges going from $L \setminus S$ to $N(p)$. Each $v \in L \setminus S$ must have at most λ_2 edges ending in $N(p)$, otherwise a biclique K_{2,λ_2+1} will be formed with p. Then the total number of edges must be bounded by,

$$|\gamma\big((L \setminus S) \cup N(p)\big)| \leq (|L| - |S|)\lambda_2 = (v-2)\lambda_2 \tag{4.4}$$

By Theorem 1, equation (2.1),

$$\lambda_1 = r = \lambda \frac{\binom{v-1}{t-1}}{\binom{k-1}{t-1}} \tag{4.5}$$

and

$$\lambda_2 = \lambda \frac{\binom{v-2}{t-2}}{\binom{k-2}{t-2}} \tag{4.6}$$

therefore combining equations (4.5) and (4.6) we have,

$$\lambda_2 = r(k-1)/(v-1) \tag{4.7}$$

By substitution of equation (4.7) in both (4.3) and (4.4) in we have,

$$|\gamma\big((L \setminus S) \cup N(p)\big)| = \frac{r(v-2)(k-1)}{v-1} + 1 > \frac{r(v-2)(k-1)}{v-1} \geq |\gamma\big((L \setminus S) \cup N(p)\big)|$$

a contradiction. □

IV.2. GDP: A novel integer programming formulation for t-designs

The equivalence result of the previous section leads to a novel linear integer programming (IP) formulation for t-design problems, which we present in this section. We use the following notation: for a graph $G = (V, E)$ and $S \subseteq E(G)$, we write $x(S) = \sum_{e \in S} x_e$.

Let x be an incidence vector of a t-design using the point-block incidence representation. Recall that $x \in \{0, 1\}^{vb}$, where v is the number of points and b is the number of blocks in the design.

Let $G = (V, E) = K_{v,b}$ be a complete bipartite graph with its vertices partitioned into two disjoint sets X and Y, where $|X| = v$ and $|Y| = b$. Since $K_{v,b}$ is a complete graph, the number of edges is $|E| = vb$.

We can establish a one-to-one relationship between the elements of E and the incidence vector x. Simply index every component of x with every edge of E and denote them x_e. These will be the decision variables of the formulation.

Let $\mathcal{K}$ be the set of all induced subgraphs $K = G[T \cup Q]$ in $G = (X \cup Y, E)$, where $T \subseteq X$, $Q \subseteq Y$, $|T| = t$, $|Q| = \lambda + 1$. The formulation GDP is finding a maximum cardinality subset of edges of E, such that:

$$
\begin{array}{|llll|}
\hline
\text{maximize } x(E) & & & \\
\text{s.t.} & x(\delta(v)) \leq r, \quad v \in X & & \text{(point-star constraints)} \\
& x(\delta(v)) \leq k, \quad v \in Y & & \text{(block-star constraints)} \\
& x(E(K)) \leq t(\lambda+1)-1, \quad K \in \mathcal{K} & & \text{(biclique constraints)} \\
& x_e \in \{0,1\}, \quad e \in E & & \\
\hline
\end{array}
\qquad \text{(GDP)}
$$

The *point star* constraints are for biclique subgraphs of the type $K_{1,b}$ and ensure that every point appears r times in all the blocks of the design. The *block star* constraints are for biclique subgraphs of the type $K_{v,1}$ and ensure that every block in the design is comprised of k points. The biclique inequalities ensure the t-balanced property of the design, i.e., that every t-set of points appear together in exactly λ blocks of the design.

As a result of the problem equivalence, if the number of blocks b used in the formulation corresponds to the parameter b number of blocks of a t-design, then the biclique inequalities will ensure that every t-set of points appear together in exactly λ blocks of the design. If fewer blocks are used in the formulation, and the point-star inequalities are eliminated, then the condition "exactly λ blocks" will be replaced by "at most λ blocks", resulting in a packing design, as defined in Section II.1. That is the same situation for a covering design, if more blocks are used and the point-star inequalities are excluded, then the condition "exactly λ blocks" will be replaced by "at least λ blocks", resulting in a covering design. The reason why the point-star inequalities are excluded, is because neither a packing nor a covering design necessarily satisfies the condition that each point is replicated exactly r times. To have so, we would need exactly b blocks as defined for a t-design.

IV.3. Polyhedral analysis

In this section, we will analyze the polyhedral aspects of GDP. We will show that the polyhedron associated with GDP is the intersection of an integral polyhedron with a non-integral polyhedron, and prove that it is not full-dimensional.

Let the right-hand side of GDP be the vector d, and $Ax \leq d$ be a system of linear inequalities in vb variables. Then the polyhedron P_{GDP} associated with GDP is a bounded polyhedron (polytope) of the form:

$$P_{GDP} = \{x \geq 0 : Ax \leq d\} \tag{4.8}$$

For simplicity of notation, let $u = t(\lambda + 1) - 1$. In matrix representation, GDP can be written as follows,

$$P_{GDP} = \{x \geq 0 : A'x \leq r, A''x \leq k, A'''x \leq u, x \leq 1\} \tag{4.9}$$

where A' is a $v \times vb$ matrix given by,

$$A' = \begin{bmatrix} 1^T & 0 & \dots & 0 \\ 0 & 1^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1^T \end{bmatrix} \tag{4.10}$$

A'' is a $b \times vb$ matrix given by,

$$A'' = \begin{bmatrix} I_b & I_b & I_b & \dots I_b \end{bmatrix} \tag{4.11}$$

where I_b is the identity matrix of size b. A''' is a $\binom{v}{t}\binom{b}{\lambda+1} \times vb$ matrix that represents all the biclique subgraphs $K_{t,\lambda+1}$ with t points and (λ+1) blocks.

In matrix form, GDP formulation looks like:

$$P_{GDP} = \{x \geq 0 : \begin{bmatrix} A' \\ A'' \\ A''' \\ I_{vb} \end{bmatrix} x \ \leq \begin{pmatrix} k \\ r \\ u \\ 1 \end{pmatrix} \} \tag{4.12}$$

In terms of polyhedra, we call $P_{biclique}$ be the polyhedron associated with the biclique inequalities and P_{star} the polyhedron associated with the star inequalities. That is, $P_{star} = \{x \geq 0 : A'x \leq r, A''x \leq k\}$ and $P_{biclique} = \{x \geq 0 : A'''x \leq u\}$. Then an alternative representation is,

$$P_{GDP} = \{x : x \in P_{star} \cap P_{biclique}, x \leq 1\} \tag{4.13}$$

A result regarding the polyhedral structure of GDP is the following:

Proposition 15 *The polyhedron P_{star} is integral.*

Proof. Let $A_s = \begin{bmatrix} A' \\ A'' \end{bmatrix}$ be the constraint matrix of the star inequalities and P_{star} be the polytope associated them. From GDP formulation, A_s is a $\{0,1\}$-matrix formed of two classes of rows: the *point star* inequalities and the *block star* inequalities. Moreover, each column of A_s contains a 1 in each of these classes (see matrix representation equations (4.10) and (4.11). Then A_s is the incidence matrix of a bipartite graph, and hence A_s is totally unimodular. Also, the right-hand-side vector is an integral vector since the parameters of designs are natural numbers by definition. Then by Hoffman and Kruskal Theorem (Theorem 12), P_{star} is integral. □

Since the constraint matrix A_s associated with the star inequalities is TU, then a stronger result that include the bounds on the variables is the following,

Proposition 16 *The polyhedron* $P_{star(\text{0-1})} = \{x \geq 0 : A'x \leq r, A''x \leq k, x \leq 1\}$ *is integral.*

Proof. The result follows directly from Hoffman and Kruskal Theorem and from the fact that also the matrix

$$\begin{bmatrix} A_s \\ I \\ -I \end{bmatrix}$$

is totally unimodular. □

It follows that another representation of GDP is

$$P_{GDP} = P_{star(\text{0-1})} \cap P_{biclique} \tag{4.14}$$

We continue with some polyhedral dimension results. By definition, the dimension of a polyhedron $P \subseteq \mathbb{R}^n$, denoted dim(P) is one less than the maximum cardinality of an affinely-independent set $X \subseteq P$ (see [17]). So, for example, to show that dim$(P) = n$, it suffices to find $n + 1$ affinely-independent points in P.

First, we examine the dimension of the biclique polyhedron $P_{biclique}$. The result is the following,

Proposition 17 *The polytope* $P_{biclique}$ *is full dimensional.*

Proof. Let the number of variables be $n = vb$. To show that $P_{biclique}$ is full dimensional, we need to find $x^1, ..., x^{n+1} \in \mathbb{R}^n$ affinely independent points in the polyhedron. Since the zero vector, $0 \in P_{biclique}$ then we need to find n linear independent points in $P_{biclique}$.

Let $x^i = (0, \ldots, 1, \ldots, 0)$ be a vector with a one in the ith position $i = 1, \ldots, n$ and zero elsewhere. Each of those points satisfy the biclique constraints. Let $x^{n+1} = 0$. Then the following linear system with $\alpha_i \in \mathbb{R}$,

$$\begin{aligned} \alpha_1 x^1 + \alpha_2 x^2 \quad &+ \ldots + \alpha_{n+1} x^{n+1} &= 0 \\ \alpha_1 + \alpha_2 \quad &+ \ldots + \alpha_{n+1} &= 0 \end{aligned}$$

has as only solution $\alpha_i = 0$ for all i. Therefore we have $n + 1$ affinely independent points. □

We continue our analysis by examining if GDP has inequalities that are satisfied as equality by all feasible points. We present a definition from polyhedral theory (see [51]).

Definition An inequality $\alpha x \leq \beta$ from $Ax \leq d$ is called an *implicit equality* if $\alpha x = \beta$ for all x satisfying $Ax \leq d$.

Another way to characterize the dimension of a polyhedron is by the following result. The dimension of a polyhedron in n variables is equal to the dimension of its affine hull. That is, equal to n minus the rank of the matrix of implicit equalities. Therefore, a polyhedron is full-dimensional if and only if there are no implicit equalities. We start by showing a result about the implicit inequalities in GDP.

We use the following notation: $A^= x \leq d^=$ is the system of implicit equalities in $Ax \leq d$, and $A^{+}x \leq d^{+}$ is the system of all other inequalities in $Ax \leq d$. We will prove that the implicit equalities of GDP are the star inequalities. We define the complement of a graph with respect to a graph that contains it as follows.

Definition Given a graph $G = (V, E)$ and a spanning subgraph of G, $H = (V, F)$, the *complement* of H with respect to G is the graph $\overline{H} = (V, E \setminus F)$.

The result about the implicit inequalities of GDP is the following.

Proposition 18 *All points $x \in P_{GDP}$ satisfy the* star *inequalities as equalities, that is, the system of implicit equalities of GDP corresponds to P_{star}.*

Proof. Let G be the bipartite graph associated with a t-design, and consider its graph complement $\overline{G}$ with respect to the complete bipartite graph $K_{v,b}$. (In terms of the point-block incidence matrix, this complement will be the matrix obtained by interchanging the 1's and 0's). It is not difficult to see that $\overline{G}$ corresponds to a t-design in which the point replication parameter is:

$$\overline{r} = b - r$$

and the block size is

$$\overline{k} = v - k$$

Let the complement of the binary decision variables in GDP be written as,

$$\overline{x}_e = 1 - x_e, \quad e \in E$$

We describe the star inequalities in GDP corresponding to $\overline{G}$. The point-star inequalities are,

$$x(E(\overline{K}_{1,b})) \leq \overline{r}$$

Taking the complement of the decision variables in the above inequality gives,

$$(1)(b) - x(E(K_{1,b})) \leq \overline{r}$$

rearranging the above,

$$x(E(K_{1,b})) \geq b - \overline{r} = r \tag{4.15}$$

Since the point-star inequalities on G are,

$$x(E(K_{1,b})) \leq r \tag{4.16}$$

by (4.15) and (4.16) we conclude that the point-star inequalities of GDP are satisfied as equalities by all feasible points. In a similar way, we consider the block-star inequities. Those are, with respect to $\overline{G}$,

$$x(E(\overline{K}_{v,1})) \leq \overline{k}$$

by complementing the decision variables in the above inequality we obtain,

$$(1)(v) - x(E(K_{v,1})) \leq \overline{k}$$

that is,

$$x(E(K_{v,1})) \geq v - \overline{k} = k \tag{4.17}$$

Since the block-star inequalities on G are,

$$x(E(K_{v,1})) \leq k \tag{4.18}$$

the result that the block-star inequalities are also implied equations follows from (4.17) and (4.18).

□

According to Proposition 18, the polytope associated with GDP can be written as:

$$P_{GDP} = \{x : A^{=}x \leq d^{=}, A^{+}x \leq d^{+}\} \tag{4.19}$$

where $A^{=}x \leq d^{=}$ are the star inequalities and $A^{+}x \leq d^{+}$ are the biclique inequalities jointly with the bounds on the variables.

In our case, by Proposition 18 there are some implicit equalities in P_{GDP}, so we can conclude that P_{GDP} is not full-dimensional. It follows that the dimension of P_{GDP} is equal to vb minus the rank of the matrix $A^{=}$, where v is the number of points in the design, and b is the number of blocks in the design. We have the following final result,

Proposition 19 *The dimension of the polytope GDP is equal to:*

$$dim(P_{GDP}) = vb - (v + b - 1)$$

Proof. According to the result proven in Proposition 18, we need to show that the rank of $A^{=}$ is $v+b-1$. Recall that $A^{=}$ is partitioned in two sets of rows, the point-star inequalities and the block-star inequalities. It is not difficult to see from the matrix representation (4.10) and (4.11) that the rows within these two classes are linearly independent. Now, any one row from the point-star inequalities can be obtained by adding the product of itself with all the rows in the block-star inequalities. The conversely is not true, since by the Fisher inequality we have $v \leq b$. Therefore $A^{=}$ is not full row rank, but its number of rows minus one. □

CHAPTER V

NEW CLASSES OF VALID INEQUALITIES FOR GDP

For hard combinatorial optimization problems, the complete description of them by means of linear inequalities is not available. The research effort in polyhedral combinatorics is to find methods to approximate this polyhedral description as good as possible so linear and integer programming algorithms would be able to solve difficult problems in a reasonable amount of time. One way to strengthen a formulation to better approximate the convex hull of integer solutions to an integer programming problem is by generating new classes of valid inequalities. We start this chapter by deriving lower bounds for the biclique inequalities in GDP. Then in Sections V.2 and V.3, we apply the known Chvátal-Gomory cut technique and give a cutting plane proof for some new classes of valid inequalities. Finally, in Section V.4 we introduce other class of valid inequalities equivalent to the star inequalities of GDP but that avoid the integrality property. As we will explain later in more detail, in a branch-and-cut computational implementation, sometimes the integrality property of the subproblem is not desirable for the fathoming criteria.

V.1. Lower bound derivation for biclique inequalities

We will derive a lower bound for the biclique inequalities in GDP. This lower bound will reduce the feasible region for some instances. Also, this lower bound will be useful in the computation of bounds for other classes of valid inequalities that will be presented in the next chapter. The derivation for the lower bound uses the concept of set of neighbors of a vertex set (as defined in Chapter IV) and the definition of λ_s given in Theorem 1. The proof also uses a fundamental rule of combinatorics called the *inclusion-exclusion principle* (see [30]).

Theorem 20 (The Inclusion-Exclusion Principle) *Let $M_1, M_2, \ldots M_n$ be finite sets. Then*

$$\begin{aligned}|M_1 \cup M_2 \cup \ldots M_n| =& |M_1| + |M_2| + \cdots + |M_n| \\ & - |M_1 \cap M_2| - |M_1 \cap M_3| - \cdots - |M_{n-1} \cap M_n| \\ & + |M_1 \cap M_2 \cap M_3| + \cdots + |M_{n-2} \cap M_{n-1} \cap M_n| \\ & \ldots \\ & + (-1)^{n+1}|M_1 \cap M_2 \cap \cdots \cap M_n|\end{aligned}$$

As explained by Herman et al. [30], on the right-hand side the rows are displayed in order such that the summands corresponding to the i-element subsets are placed in the ith row. Then row i contains exactly $\binom{n}{i}$ summands. The total number of summands on the right-hand side is therefore $2^n - 1$.

Let $G = (X \cup Y, E)$ be a bipartite graph and let $\mathcal{K}$ be the set of all induced subgraphs $K = G[T \cup Q]$ in $G = (X \cup Y, E)$, where $T \subseteq X$, $Q \subseteq Y$, $|T| = t$, $|Q| = \lambda + 1$. For any t-set of vertices $T \subseteq X$, we write its neighbors as $N(T) \subseteq Y$. Our result regarding the lower bound for the biclique inequalities is the following,

Proposition 21 *For any $K \in \mathcal{K}$,*

$$x(E(K)) \geq \textit{max}\, \{0, \lambda + 1 - b + |N(T)|\} \tag{5.1}$$

where

$$|N(T)| = \sum_{s=1}^{t} (-1)^{s+1} \binom{t}{s} \lambda_s \tag{5.2}$$

Proof. For any t-set of vertices $T \subseteq X$, we will show the cardinality of $N(T)$. By definition of a t-design, every s-set of X for $0 \le s \le t$ appear together in λ_s blocks. So all s-sets intersections have the same cardinality for $0 \le s \le t$. The number of sets of size s out of a t set is given by the number of combinations $\binom{t}{s}$. Then the right-hand side of Theorem 20 reduces to,

$$\binom{t}{1}\lambda_1 - \binom{t}{2}\lambda_2 + \binom{t}{3}\lambda_3 + \cdots + (-1)^{t+1}\binom{t}{t}\lambda_t$$

where $\lambda_t = \lambda$ for a t-design. The set of neighbors of T is the union of the neighbors of each point $v \in T$, therefore the above can be written as,

$$|N(T)| = \sum_{s=1}^{t}(-1)^{s+1}\binom{t}{s}\lambda_s \tag{5.3}$$

Now, to show the lower bound for the biclique inequality we need to look at the complement $Y \setminus N(T)$ (see Figure 5 for an example). The number of vertices in this complement is,

$$|Y \setminus N(T)| = b - |N(T)| \tag{5.4}$$

Since the biclique inequalities are for all induced $K = G[T \cup Q]$ subgraphs of G, consider in particular a K with point-vertices equal to T, and with block-vertices Q intersecting $Y \setminus N(T)$. Then K requires $|Q| = \lambda + 1$ block-vertices, and there are available (5.4) outside the neighborhood $N(T)$, therefore the difference between the required and the available will provide the lower bound as follows,

$$|Q \cap N(T)| \ge (\lambda + 1) - (b - |N(T)|) \tag{5.5}$$

Since T was chosen arbitrarily, and the edge count must be nonnegative, the result follows. □

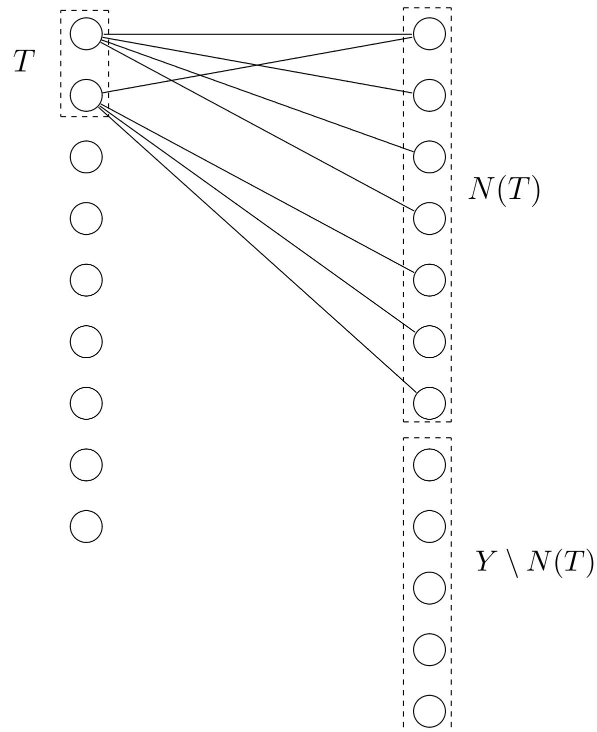

Fig. 5. Illustration for the lower bound proof.

V.2. Chvátal-Gomory cuts

We derive new classes of valid inequalities or cuts for our formulation GDP using the biclique inequalities. A general method to generate valid inequalities for all integral vectors in a polyhedron is the *Chvátal-Gomory cutting plane method*. These inequalities are also called *cutting planes* or *cuts*. Here we include a description as found in Cook et al. [17].

Given a system of m linear inequalities of the form

$$a_i^T x \leq b_i \qquad (i = 1, \ldots, m) \tag{5.6}$$

Let $y_1, \ldots, y_m$ be nonnegative real numbers and set

$$c = \sum (y_i a_i : i = 1, \ldots, m) \tag{5.7}$$

and

$$d = \sum (y_i b_i : i = 1, \ldots, m). \tag{5.8}$$

Every solution to (5.6) satisfies $c^T x \leq d$. Moreover, if c is integral, then all integral solutions to (5.6) also satisfy the stronger inequality

$$c^T x \leq \lfloor d \rfloor \tag{C-G cut}$$

where $\lfloor d \rfloor$ denotes d rounded down to the nearest integer. Inequality (C-G cut) is called a *Chvátal-Gomory cut*. Once a cut is derived, it can be used to derive further inequalities. A sequence of such derivations is called a *cutting plane proof*.

In the case of our formulation GDP, let $K_{v,b} = G = (V \cup B, E)$ be the complete bipartite graph in which the formulation is based. We call *points* the vertices belonging to the partition V and *blocks* the vertices belonging to the partition B of G. Note that any two $K_{t,\lambda+1}$ biclique subgraphs of G can have their corresponding vertex set completely disjoint or have some common vertices up to differing in only one vertex (either a point or a block vertex). This implies that some bicliques are edge-disjoint and others have in common one or more edges (see Figure 6 for an example). A pair of biclique graphs with maximum number of common edges can be classified in one of the following two cases:

Case 1. The node set of two bicliques differ by exactly one node in the point partition. Then we have $(t-1)(\lambda+1)$ common edges.

Case 2. The node set of two bicliques differ by exactly one node in the block par-

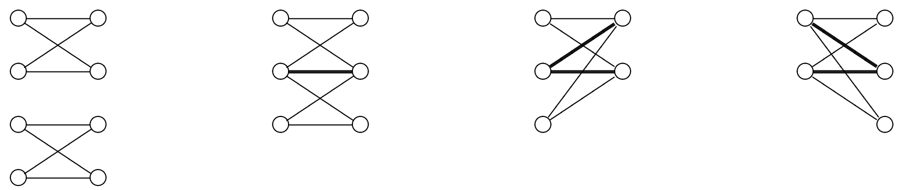

Fig. 6. Examples of two bicliques $K_{2,2}$ that (a) are edge-disjoint, (b) share one edge, (c) share two edges as in case 1, (d) share two edges as in case 2.

tition. Then we have $t(\lambda + 1 - 1)$ common edges.

From the above, the maximum number of common edges in any two bicliques is $\max\{(t-1)(\lambda+1), \lambda t\}$. In terms of the biclique inequalities in GDP, this value gives the maximum number of common variables in any two such inequalities. The cut derivation procedure is the same for both cases, only the resulting biclique associated with the cut will be different as will be explained later.

Consider the set of m biclique inequalities from GDP written as in (5.6). Let p and q be indices from $i = 1, \ldots, m$ of two inequalities with maximum number of common variables. Let $G_p = (V_p, E_p)$ and $G_q = (V_q, E_q)$ be the biclique graphs corresponding to inequalities p and q, respectively. Let $y_p = y_q = 1$ and $y_i = 0$ for all $i = 1, \ldots, m$ such that $i \neq p, q$. Recall that all the biclique inequalities coefficients are equal to one, and that the right-hand side is a constant.

Denote the union graph $G_\cup = G_p \cup G_q = (V_p \cup V_q, E_p \cup E_q)$, the intersection graph $G_\cap = G_p \cap G_q = (V_p \cap V_q, E_p \cap E_q)$ (see [20]). Compute $c = y_p a_p + y_q a_q = a_p + a_q$ and $d = y_p b_p + y_q b_q = b_p + b_q = 2(t(\lambda+1) - 1)$. Using the notation $x(E) = \sum_{e \in E} x_e$ then,

$$c^T x = (a_p + a_q)^T x = x(E_p) + x(E_q) = x(E_\cup) + x(E_\cap)$$

Since G_p and G_q are two bicliques $K_{t,\lambda+1}$ with maximum number of common edges, the intersection graph will be either $G_\cap = K_{t-1,\lambda+1}$ for case 1 or $G_\cap = K_{t,\lambda}$ for case 2. Let Z be the point vertices of $G_\cap$ for case 1, or the block vertices of $G_\cap$ for case 2. Let $G_\triangle = (V_\triangle, E_\triangle)$ be the induced subgraph $G_\cup \backslash Z$. It is not difficult to see that $x(E_\cup) = x(E_\triangle) + x(E_\cap)$. Then,

$$c^T x = x(E_\cup) + x(E_\cap) = x(E_\cup) + x(E_\cup) - x(E_\triangle) = 2x(E_\cup) - x(E_\triangle)$$

Note that biclique $G_\triangle$ is of the same size as the other bicliques G_p and G_q, so its corresponding inequality belongs also to the system (5.6). That is, $x(E_\triangle) \leq t(\lambda + 1) - 1$ is valid. Therefore, from equation (C-G cut), a new class of Chvátal-Gomory valid inequalities for GDP is:

$$x(E_\cup) \leq \left\lfloor \frac{3(t(\lambda+1)-1)}{2} \right\rfloor \qquad \text{(New Class 1)}$$

where $E_\cup$ is the edge set of the union graph $G_\cup$ of two bicliques $K_{t,\lambda+1}$ that differ in only one vertex. Note that for this class, $G_\cup = K_{t+1,\lambda+1}$ for case 1 and $G_\cup = K_{t,\lambda+2}$ for case 2.

For example, if $t = 2$ and $\lambda = 1$, consider the two $K_{2,2}$ bicliques shown in Figure 7: $G_p = (V_p, E_p) = (\{1,2,4,5\}, \{14,15,24,25\})$ and $G_q = (V_q, E_q) = (\{2,3,4,5\}, \{24,25,34,35\})$. Those bicliques have maximum number of common edges as in case 1. The union $G_\cup$, intersection $G_\cap$, and symmetric difference $G_\triangle$ graphs are also depicted. The two corresponding inequalities for bicliques p and q are,

$$x_{14} + x_{15} + x_{24} + x_{25} \leq 3$$

$$x_{24} + x_{25} + x_{34} + x_{35} \leq 3$$

adding them we get,

$$x_{14} + x_{15} + 2x_{24} + 2x_{25} + x_{34} + x_{35} \leq 6$$

that is,

$$2(x_{14} + x_{15} + x_{24} + x_{25} + x_{34} + x_{35}) - (x_{14} + x_{15} + x_{34} + x_{35}) \leq 6$$

but $(x_{14} + x_{15} + x_{34} + x_{35}) \leq 3$, since the inequality corresponds to other biclique, $G_{\triangle}$, in the system. Therefore,

$$x_{14} + x_{15} + x_{24} + x_{25} + x_{34} + x_{35} \leq \left\lfloor \frac{6+3}{2} \right\rfloor = 4$$

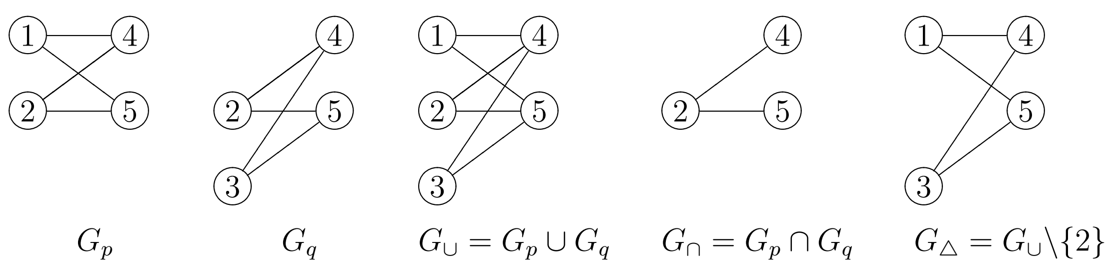

Fig. 7. Example of initial Chvátal-Gomory cut generation for GDP.

V.3. Cutting plane proof

A second class of Chvátal-Gomory cuts can be derived in the same way using the original biclique inequalities and the new generated class of inequalities. Let G_p be a biclique $K_{t,\lambda+1}$ from (GDP) and G_r be a biclique from (New Class 1) with maximum number of common edges. For case 1, $G_r = K_{t+1,\lambda+1}$; for case 2, $G_r =$

$K_{t,\lambda+2}$. For simplicity, let $u_0 = t(\lambda+1) - 1$ and $u_1 = \lfloor \frac{2u_0+u_0}{2} \rfloor$. Let $y_p = y_r = 1$, and zero for the rest of the indices in equations (5.7) and (5.8). Compute $c = y_p a_p + y_r a_r = a_p + a_r$, and $d = y_p u_0 + y_r u_1 = u_0 + u_1$. The biclique graph $G_\triangle$ is created as previously. The cut is as follows,

$$c^T x = x(E_p) + x(E_r) = 2x(E_\cup) - x(E_\triangle) \leq u_0 + u_1$$

since $x(E_\triangle) \leq u_1$, a second new class of Chvátal-Gomory valid inequalities for (GDP) is:

$$x(E_\cup) \leq \lfloor \frac{2u_1 + u_0}{2} \rfloor \qquad \text{(New Class 2)}$$

where $E_\cup$ is the edge set of the union graph $G_\cup$ of a biclique $K_{t,\lambda+1}$ and biclique G_r that differ in only one vertex. Note that for this second class, $G_\cup = K_{t+2,\lambda+1}$ for case 1 and $G_\cup = K_{t,\lambda+3}$ for case 2.

A third class of valid inequalities can be derived using an inequality from the second class and an inequality from the original biclique constraints using a similar procedure. Then a fourth class can be derived, and so on. It is important to note two things. First, that these inequalities become redundant with the star inequalities in GDP at least after certain class $\bar{n}$. This $\bar{n}$ depends on the parameters t-(v,k,λ) and r of the design and is given by:

$$\bar{n} = \begin{cases} v - 2t & \text{for case 1} \\ b - 2(\lambda+1) & \text{for case 2} \end{cases} \tag{5.9}$$

Second, that the fact that a block is comprised of k points, and that a point is replicated r times in a t-design provides a global upper bound in both cases. That is, the number of edges in a biclique subgraph is at most:

$$\bar{u} = \begin{cases} k(\lambda+1) & \text{for case 1} \\ tr & \text{for case 2} \end{cases} \tag{5.10}$$

Table 11 summarizes the upper bound results of this Chvátal-Gomory cutting plane proof. The biclique case with larger number of edges will yield stronger inequalities since the upper bounds are the same. The stronger inequality is called an *extended* or *lifted* inequality in combinatorial optimization terms.

Table 11. Summary of upper bounds for new classes of valid inequalities derived by Chvátal-Gomory cutting plane proof

Class	$V(G_\cup)$ (case 1)	$V(G_\cup)$ (case 2)	C-G derived upper bound	C-G cut: $x(E_\cup) \le u_n$ final upper bound
0	$K_{t,\lambda+1}$	$K_{t,\lambda+1}$	$t(\lambda+1)-1 = u_0$	$u_0 = \min\{\bar{u}, u_0\}$
1	$K_{t+1,\lambda+1}$	$K_{t,\lambda+2}$	$\lfloor(2u_0+u_0)/2\rfloor = u_1$	$u_1 = \min\{\bar{u}, u_1\}$
2	$K_{t+2,\lambda+1}$	$K_{t,\lambda+3}$	$\lfloor(2u_1+u_0)/2\rfloor = u_2$	$u_2 = \min\{\bar{u}, u_2\}$
$\vdots$				$\vdots$
n	$K_{t+n,\lambda+1}$	$K_{t,\lambda+1+n}$	$\lfloor(2u_{n-1}+u_0)/2\rfloor = u_n$	$u_n = \min\{\bar{u}, u_n\}$
$\vdots$				$\vdots$
$\bar{n}$	$K_{v-t,\lambda+1}$	$K_{t,b-(\lambda+1)}$	$\lfloor(2u_{\bar{n}-1}+u_0)/2\rfloor = u_{\bar{n}}$	$u_{\bar{n}} = \bar{u}$

V.4. Valid inequalities that avoid the integrality property

The following procedure allows us to find the last biclique inequality which is not redundant with P_{star} (facet-defining for the factor polyhedron) and yet as strong as possible.

Consider the complete bipartite graph $G = K_{v,b}$ and a biclique-free factor solution F on it that corresponds to a design. If we delete one vertex for the block partition and one vertex for the point partition and their corresponding edges on F we obtain a biclique-free factor solution F_1 on $K_{v-1,b-1}$. The number of edges deleted is at most $k + r$ and at least $k + r - 1$. Denote by $\mathcal{D}$ the class of all $K_{v-1,b-1}$ biclique subgraphs of G. Then the class of inequalities that we call *Maximal fractional star class* defined as follows,

$$bk - (k + r) \leq x(E(D)) \leq bk - (k + r) + 1, \quad D \in \mathcal{D} \tag{5.11}$$

is a class of valid inequality for GDP.

This class of inequalities can be used in GDP in lieu of the star inequalities to avoid the integrality property in the branch-and-cut search. Wilhelm [60] observes the issues regarding the integrality property in a column generation context. Here we translate the same idea in a cutting plane and branch-and-cut context. If a submodel has the integrality property, all of its extreme points are integer solutions so that the LP algorithm can solve it quickly. But the disadvantage is that the LP bounds obtained after adding the violated cuts will not be tighter than the LP relaxation of the original IP. As pointed out by Wilhelm [60], "this will not facilitate branch-and-bound fathoming and overall run time may, therefore, be prohibitive".

Table 12 gives an example of the LP relaxation bounds obtained analytically for a 2-(8,4,3) design. Note that the bounds are very close to the optimal and that the optimal extreme points are non-integral.

Table 12. LP bounds for maximal fractional star class for design 2-(8,4,3)

Class	Vars in cut	Total vars	Sense	LP Bound	Optimal z_{LP}
$\mathcal{D}$	(7)(13) = 91	(8)(14) = 112	min	112(45/91)=55.39	(4)(14)= 56
$\mathcal{D}$	91	112	max	112(46/91)=56.61	(4)(14)= 56

The class 5.11 of inequalities ensures the block size and point replication of the solution. To ensure the t-balanced property, we need either the biclique inequalities or a class derived from them, which is the topic of the next chapter.

CHAPTER VI

STRONGER VALID INEQUALITIES FOR GDP BY COMPLEMENT AND SUPPLEMENT

The Chvátal-Gomory classes of valid inequalities derived for GDP in the previous chapter become weaker for t-designs with $\lambda > 1$ as the number of variables in the cuts increases. In this chapter we address the following questions as Wolsey [63] formulated for another problem, but applied to our problem: Are the biclique inequalities in GDP strong? Is it possible to strengthen the biclique inequalities so that they provide better cuts? The purpose of having cuts as strong as possible is that they provide better LP relaxation bounds. These better bounds help reduce the size of the enumeration tree in a branch-and-bound search.

In Section VI.1, we describe how we can complement induced subgraphs and complement binary variables to obtain a stronger version of the biclique inequalities. In this procedure, we use both the original upper bound for the biclique inequalities and the lower bound derived in the previous chapter to derive three classes of valid inequalities. We also give an analysis of the strength of the linear programming relaxation bounds to compare these classes. In Section VI.2 we utilize the concept of supplementary incidence structures of design theory to generate other equally strong class of valid inequalities which we call *stable-set class*. A *stable set* (or *independent set*) is a set of vertices in a graph that are pairwise disjoint. The stable-set class is particularly interesting because its lower bound is a constant, independent of the parameters of a t-(v, k, λ) design. Other three classes derived from the stable-set class are also given. We claim that any of these seven new classes of valid inequalities are equivalent, and can be used in lieu of the original biclique inequalities of GDP to construct a t-design. However, the polyhedra associated with them are not the same.

VI.1. Valid inequalities by graph complement

Here we use again the notation of induced subgraph as introduced in Section IV.1 on page 43. We start with a bipartite graph $G = (X \cup Y, E)$ and a induced subgraph $K = G[T \cup Q]$ as defined for the class of biclique inequalities in GDP. Define the complement of K with respect to G as $K' = [T' \cup Q']$, where $T' = X \setminus T$ and $Q' = Y \setminus Q$. The bounds for the classes obtained by these type of graph complements will be derived from both the upper and lower bound of the original biclique class of inequalities.

VI.1.1. From original biclique class

With the above definition of complement of K with respect to G, we have a partitioned of the vertex set of G into four mutually disjoint sets of vertices T, Q, T' and Q', each of cardinality t, $\lambda+1$, $v-t$ and $b-\lambda-1$, respectively (see Figure 8 for an example). Recall the notation for a graph $G = (V, E)$ and $A \subseteq V$, $\delta(A) = \{e \in E :$ e has an end in A and an end in $V \setminus A\}$, $\gamma(A) = \{e \in E :$ both ends of e are in $A\}$. Using this notation, a biclique inequality for an induced subgraph K in GDP is then written:

$$x(\gamma(T \cup Q)) \leq t(\lambda+1) - 1 = u_0 \tag{6.1}$$

For a t-(v,k,λ) design, by definition, every block is of size k,

$$x(\delta(Q)) = k|Q| = k(\lambda+1) \tag{6.2}$$

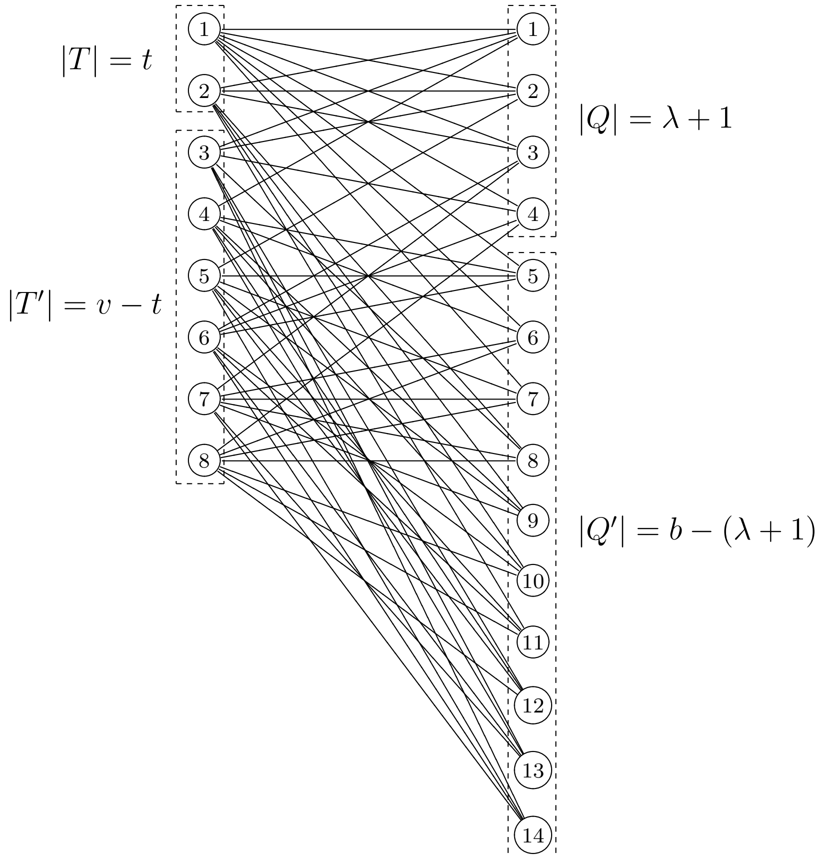

Fig. 8. Illustration for the derivation of graph complement classes.

By the complement of the point set of the graph K, it follows that,

$$x(\delta(Q)) = x(\gamma(T \cup Q)) + x(\gamma(T' \cup Q)) \tag{6.3}$$

Combining (6.1), (6.2) and (6.3) we have,

$$x(\gamma(T' \cup Q)) \geq k(\lambda + 1) - u_0 \tag{6.4}$$

In the same way, for a t-design with point replication r, it is true by definition that,

$$x(\delta(T)) = r|T| = rt \tag{6.5}$$

By complementing the block set of the graph K, it follows that,

$$x(\delta(T)) = x(\gamma(T \cup Q)) + x(\gamma(T \cup Q')) \tag{6.6}$$

Combining (6.1), (6.5) and (6.6) we have,

$$x(\gamma(T \cup Q')) \geq rt - u_0 \tag{6.7}$$

We still have another complement of K which is the vertex-disjoint complement K'. Again, for a t-design it is true by definition that,

$$x(\delta(Q')) = k|Q'| = k(b - (\lambda + 1)) \tag{6.8}$$

By complement it follows that,

$$x(\delta(Q')) = x(\gamma(T \cup Q')) + x(\gamma(T' \cup Q')) \tag{6.9}$$

Combining (6.8), (6.9) and (6.7) we have,

$$x(\gamma(T' \cup Q')) \leq k(b - (\lambda + 1)) - (rt - u_0) \tag{6.10}$$

The same bound obtained in (6.10) could have been obtained by using $\delta(T')$ in (6.8) instead of $\delta(Q')$ as follows,

$$x(\delta(T')) = r|T'| = r(v - t) \tag{6.11}$$

By complement of the block-vertex set it follows that,

$$x(\delta(T')) = x(\gamma(T' \cup Q)) + x(\gamma(T' \cup Q')) \tag{6.12}$$

Combining (6.11), (6.12) and (6.4) we have,

$$x(\gamma(T' \cup Q')) \leq r(v-t) - (k(\lambda+1) - u_0) \tag{6.13}$$

The proof that the bound given by both equations (6.10) and (6.13) is the same comes from the fact that for a t-design it is true by definition that $vr = bk$. Table 13 summarizes the results of the four equivalent classes of biclique inequalities obtained. It is worth to note that the bounds in Table 13 are exact in the sense that at least one biclique in the class will satisfy its corresponding inequality as equality.

Table 13. Stronger classes of inequalities by complementing from biclique class

Class	$V(K)$	Class size	Bound for $\lvert E(K)\rvert$
0	$K_{t,\lambda+1}$	$\binom{v}{t}\binom{b}{\lambda+1}$	upper u_0: $t(\lambda+1)-1$
0′	$K_{v-t,\lambda+1}$	$\binom{v}{t}\binom{b}{\lambda+1}$	lower l_0': $(k-t)(\lambda+1)+1$
0″	$K_{t,b-(\lambda+1)}$	$\binom{v}{t}\binom{b}{\lambda+1}$	lower l_0'': $t(r-\lambda-1)+1$
0‴	$K_{v-t,b-(\lambda+1)}$	$\binom{v}{t}\binom{b}{\lambda+1}$	upper u_0''': $r(v-t)-(k-t)(\lambda+1)-1$

For example, consider a design 2-(8,4,3) with $b = 14$, $r = 7$. The bounds are shown in Table 14.

Table 14. Example of stronger classes by complementing for design 2-(8,4,3)

Class	$V(K)$	Class size	Bound for $\lvert E(K)\rvert$
0	$K_{2,4}$	28028	upper: u_0=7
0′	$K_{6,4}$	28028	lower: l_0'=9
0″	$K_{2,10}$	28028	lower: l_0''=7
0‴	$K_{6,10}$	28028	upper: u_0'''=33

VI.1.2. LP relaxation bounds analysis

Despite the fact that any of the four classes in Table 13 can be used in GDP equivalently in place of the biclique inequalites, not all the classes will give the same LP relaxation solution. We are interested in the strongest biclique class since it will give better bounds for the linear programming (LP) relaxations when using a branch-and-cut algorithm, has an integral polyhedron like P_{star} be excluded from the base formulation. Here we introduce the concept of *better formulation* as treated by Wolsey [63]. First, the definition of a formulation applied to integer programming.

Definition A polyhedron $P \subseteq \mathbb{R}^n$ is a *formulation* for a set $X \subseteq \mathbb{Z}^n$ if and only if $X = P \cap \mathbb{Z}^n$.

The above definition implies that a problem may have different formulations. How can we say that a formulation is "better" than another? In [63] the following definition is presented

Definition Given a set $X \subseteq \mathbb{R}^n$ and two formulations P_1 and P_2 for X, P_1 is a *better formulation* than P_2 if $P_1 \subset P_2$.

We include and important result, the Duality Theorem of Linear Programming (see [17]),

Theorem 22 (Duality Theorem) *Let A be a $m \times n$ matrix, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. Then*

$$max\{c^T x : Ax \leq b\} = min\{y^T b : y \geq 0, y^T A = c^T\} \tag{6.14}$$

provided that both sets are nonempty

Let $P_{biclique(0)}$ be the polyhedron associated with the biclique inequalities (class 0), that is

$$P_{biclique(0)} = \{x > 0 : A_0 x \leq u_0\} \tag{6.15}$$

and let $P_{biclique(i)}$ be the polyhedron associated with complement class i ($i = 1, 2, 3$) as given in Table 13. For example, for the third class (Class 0‴),

$$P_{biclique(3)} = \{x > 0 : A_3 x \leq u_0'''\} \tag{6.16}$$

Our strength result is the following,

Proposition 23

$$P_{biclique(3)} \subset P_{biclique(0)}$$

Proof. Consider the points in $\mathbb{R}^{vb}$, $x_0 = \frac{u_0}{t(\lambda+1)}$ and $x_3 = \frac{u_0'''}{(v-t)(b-(\lambda+1))}$. Note that $x_0 \in P_0$ since $t(\lambda+1)x_0 \leq u_0$, and that $x_3 \in P_3$ since $(v-t)(b-\lambda-1)x_3 \leq u_0'''$. It is not difficult to verify that the vectors in $\mathbb{R}^{\binom{v}{t}\binom{b}{\lambda+1}}$, $y_0 \geq 0$ and $y_3 \geq 0$, defined as $y_0 = \frac{vb}{t(\lambda+1)\binom{v}{t}\binom{b}{\lambda+1}}$, $y_3 = \frac{vb}{(v-t)(b-(\lambda+1))\binom{v}{t}\binom{b}{\lambda+1}}$ are feasible solution to the dual system

$$y \geq 0, y^T A = 1^T$$

for both A_0 and A_3, respectively. The value $1^T x_0 = y_0^T u_0$, and the value $1^T x_3 = y_3^T u_3$. Therefore, by Duality Theorem 22, both x_0 and x_3 are extreme maximum points, for $P_{biclique(0)}$ and $P_{biclique(3)}$, respectively.

First show $P_{biclique(3)} \subseteq P_{biclique(0)}$. The number of variables in each constraint of $P_{biclique(0)}$ is $t(\lambda+1)$. Then $x_3 \in P_{biclique(0)}$ since

$$t(\lambda+1)x_3 \leq u_0$$

Now, show that $P_{biclique(3)}$ is strictly contained in $P_{biclique(0)}$. Consider the point $x_0 \in P_0$. The number of variables in each constraint of $P_{biclique(3)}$ is $(v-t)(b-(\lambda+1))$. Then $x_0 \notin P_3$ since,

$$x_0(v-t)(b-(\lambda+1)) = \frac{u_0(v-t)(b-(\lambda+1)}{t(\lambda+1)} = u_0\frac{b(v-t)-v(\lambda+1)}{t(\lambda+1)} + u_0 \geq u_0$$

□

Consider again the example for a 2-(8,4,3) design given in Table 14. In this example, $x_0 \in \mathbb{R}^{112} = (7/8)\mathbf{1}$ and $x_3 \in \mathbb{R}^{112} = (33/60)\mathbf{1}$. The dual vectors are $y_0 \in \mathbb{R}^{28028} = (1/2002)\mathbf{1}$ and $y_3 \in \mathbb{R}^{28028} = (1/15015)\mathbf{1}$. The objective function for Class 0 is $1^T x_0 = 112(7/8) = 98 = y_0^T u_0 = 280280(1/2002)(7)$. The objective function for the Class 0‴ is $1^T x_3 = 112(33/60) = 61.6 = y_3^T u_0''' = 28028(1/15015)(33)$. The point x_3 satisfies the biclique inequalities of Class 0, $x(E(K_{2,4}) \leq 7$; while the point x_0 does not satisfy the inequalities of Class 0‴, $x(E(K_{6,10})) \leq 33$.

We conclude that a better formulation, as defined by Wolsey [63], among the four complement classes summarized in Table 13, is the third complement class, Class 0‴. We conjecture that this class dominates the biclique inequalities in GDP and will therefore give better bounds for the LP relaxations when using a branch-and-cut algorithm if the base formulation does not have the integrality property (see section V.4 for more about the integrality property). An example of the LP bounds for maximization for 2-(8,4,3) is given in Table 15.

VI.1.3. From biclique class lower bound

We complete the bounds for the complement classes by using the lower bound derived in Proposition 21 on page 60. The procedure is exactly the same as with the other bounds. Given a bipartite graph $G = (X \cup Y, E)$ and any induced subgraph

Table 15. LP bounds for biclique and complementing class for design 2-(8,4,3)

Class	Variables in cut	Total variables	LP relaxation bound z_{LP}	Optimal z^*
0	(2)(4) = 8	(8)(14) = 112	112(7/8)=98	(4)(14)= 56
0‴	(6)(10) = 60	112	112(33/60)=61.6	56

$K = G[T \cup Q]$ of G, as defined for the class of biclique inequalities $\mathcal{K}$ of GDP. The number of edges with one end in T and the other end in Q is given by (5.1). For simplicity of notation we call this bound l_0. That is,

$$x(E(K)) \geq \max\{0, \lambda + 1 - b + |N(T)|\} = l_0$$

Taking the complement of the biclique K with respect to the point-partition, X, we have that the number of edges with one end in $X \setminus T$ and the other end in Q is bounded above by

$$x(\gamma((X \setminus T) \cup Q) \leq k(\lambda + 1) - l_0 = u_0' \tag{6.17}$$

Now taking the complement of the biclique K with respect to the block-partition, Y, we have that the number of edges with one end in T and the other end in $Y \setminus Q$ is bounded above by

$$x(\gamma(T \cup (Y \setminus Q))) \leq tr - l_0 = u_0'' \tag{6.18}$$

For the last graph complement, the number of edges that have one end in $Y \setminus Q$ can be expressed as the following sum,

$$x(\delta(Y \setminus Q)) = x(\gamma(T \cup (Y \setminus Q))) + x(\gamma((X \setminus T) \cup (Y \setminus Q)) \tag{6.19}$$

Also, the number of edges that have one end in $Y \setminus Q$ is,

$$x(\delta(Y \setminus Q)) = k(b - (\lambda + 1)) \tag{6.20}$$

Combining equations (6.18), (6.19), and (6.20) we have,

$$x(\gamma((X \setminus T) \cup (Y \setminus Q)) \geq k(b - (\lambda + 1)) - u_0'' \tag{6.21}$$

which is equivalent to

$$x(\gamma((X \setminus T) \cup (Y \setminus Q)) \geq r(v - t) - u_0' \tag{6.22}$$

Table 16 summarizes these results, and Table 17 shows an example for a design 2-(8,4,3) with $b = 14$, $r = 7$.

Table 16. Stronger classes of inequalities by complementing from biclique class lower bound

Class	$V(K)$	Class size	Bound for $\|E(K)\|$
0	$K_{t,\lambda+1}$	$\binom{v}{t}\binom{b}{\lambda+1}$	lower l_0: max$\{0, \lambda + 1 - b + \|N(T)\|\}$
0′	$K_{v-t,\lambda+1}$	$\binom{v}{t}\binom{b}{\lambda+1}$	upper u_0': $k(\lambda + 1) - l_0$
0″	$K_{t,b-(\lambda+1)}$	$\binom{v}{t}\binom{b}{\lambda+1}$	upper u_0'': $tr - l_0$
0‴	$K_{v-t,b-(\lambda+1)}$	$\binom{v}{t}\binom{b}{\lambda+1}$	lower l_0''': $k(b - \lambda - 1) - tr + l_0$

Table 17. Example of stronger classes by complementing from lower bound for design 2-(8,4,3)

Class	$V(K)$	Class size	Bound for $\|E(K)\|$
0	$K_{2,4}$	28028	lower: l_0=1
0′	$K_{6,4}$	28028	upper: u_0'=15
0″	$K_{2,10}$	28028	upper: u_0''=13
0‴	$K_{6,10}$	28028	lower: l_0'''=27

VI.2. Stable-set class of valid inequalities

In the previous section we derived stronger classes of valid inequalities by graph complementing the original biclique inequalites for GDP. Now we present other strong classes of valid inequalities derived form the original biclique inequalities by supplementing the incidence structure. The following definition from design theory can be found in Beth et al. [8]. There the term *complementary* structure is used instead of *supplementary*. We prefer the second to be consistent with the terminology of Kreher [37].

Definition The *supplementary* structure of an incidence structure D=(X,$\mathcal{B}$,I) is the incidence structure $\overline{D}$=(X,$\mathcal{B}$,J) with $J = (X \times \mathcal{B})\backslash I$.

If D is the point-block incidence matrix of a t-design, then incidence matrix $\overline{D}$ of the supplementary structure is obtained by interchanging the 0's and 1's in D. In terms of the decision variables in GDP, we define the new supplementing variables as,

$$\overline{x}_e = 1 - x_e, \quad e \in E \tag{6.23}$$

If each point is on r blocks and if any two points x,y are on exactly λ blocks of D, then any two points are on exactly $(b - 2r + \lambda)$ blocks of $\overline{D}$ [8]. A generalization for t-designs is the following Theorem found in Kreher [37],

Theorem 24 (see [37]) *For a t-(v,k,λ) design and any s-set S of points with $0 \leq s \leq t$, the number of blocks that do not contain any point of S is*

$$\overline{\lambda}_s = |\{B \in \mathcal{B} : B \cap S = \emptyset\}| = \frac{\lambda\binom{v-s}{k}}{\binom{v-t}{k-t}}, \qquad 0 \leq s \leq t \tag{6.24}$$

In particular, for $s = t$ we denote $\overline{\lambda}_t = \overline{\lambda}$. Then the supplement of a t-(v,k,λ) design is a t-(v,v-k,$\overline{\lambda}$) design. The point replication parameter for the supplement design is $\overline{r} = b - r = \overline{\lambda}_1$.

The above can be used to generate a new class of valid inequalities for GDP for t-designs as follows. Let $\overline{G} = (X \cup Y, \overline{E})$ be a bipartite graph associated with the supplement structure $\overline{D}$. Let $\overline{K}$ be an induced subgraph of $\overline{G}$, that is, $\overline{K} = \overline{G}[T, Q]$ where $T \subseteq X$ and $Q \subseteq Y$, $|T| = t$, $|Q| = \overline{\lambda} + 1$. The corresponding biclique inequality is:

$$x(E(\overline{K})) \leq t(\overline{\lambda} + 1) - 1 \tag{6.25}$$

We define K, the complement of the graph $\overline{K}$ with respect to the complete bipartite graph $K_{t,\overline{\lambda}+1}$, to be the graph on vertices $T \cup Q$ with the edges in $K_{t,\overline{\lambda}+1}$ but not in $\overline{K}$. Taking supplementing variables as defined in (6.23), the above equation can be written as

$$t(\overline{\lambda} + 1) - x(E(K)) \leq t(\overline{\lambda} + 1) - 1 \tag{6.26}$$

that is,

$$x(E(K)) \geq 1 \qquad \text{(Stable-set Class, or Class } \bar{0}\text{)}$$

In the same way that from the original biclique inequalities (Class 0) other classes of valid inequalities were derived by graph complements, this stable-set class (Class $\bar{0}$) can also be used to generate other classes of valid inequalities. Table 18 summarizes the results of these new classes of biclique inequalities obtained. The bounds given are with respect to the original incidence structure of GDP, and were obtained from the bounds in the supplement incidence structure by supplementing variables as in equation (6.26). Again, these bounds proved exact in the sense that at least one subgraph in the class satisfy the bounds at equality.

Table 18. Stronger classes of inequalities from stable-set class

Class	$V(K)$	Class size	Bound for $\|E(K)\|$
$\overline{0}$	$K_{t,\overline{\lambda}+1}$	$\binom{v}{t}\binom{b}{\overline{\lambda}+1}$	lower: 1
$\overline{0}'$	$K_{v-t,\overline{\lambda}+1}$	$\binom{v}{t}\binom{b}{\overline{\lambda}+1}$	upper: $k(\overline{\lambda}+1)-1$
$\overline{0}''$	$K_{t,b-(\overline{\lambda}+1)}$	$\binom{v}{t}\binom{b}{\overline{\lambda}+1}$	upper: $rt-1$
$\overline{0}'''$	$K_{v-t,b-(\overline{\lambda}+1)}$	$\binom{v}{t}\binom{b}{\overline{\lambda}+1}$	lower: $r(v-t)-k(\overline{\lambda}+1)+1$

As an example, consider the design 2-(7,3,2) with $b = 14$ blocks and $r = 6$, represented by the bipartite graph $G = (X \cup Y, E)$. Compute $\overline{\lambda}$=4 using Theorem 24 with $s = t$; compute $\overline{r} = 8$ and $\overline{k} = 8$. Consider $T \subseteq X$ and $Q \subseteq Y$ and define the complements $T' = X \setminus T$ and $Q' = Y \setminus Q$. We denote by $\overline{\gamma}(V)$ a set of edges in the supplement structure induced by the vertices V.

Consider first the case $|T| = t$ and $|Q| = \overline{\lambda} + 1$. By the biclique inequalites in GDP on the supplementary incidence structure of the design we have,

$$x(E(\overline{K})) = x(\overline{\gamma}(T \cup Q)) \leq (2)(5) - 1 = 9, \quad \text{where } |T| = 2, |Q| = 5 \qquad (6.27)$$

Take an arbitrary $T \subseteq X$, $|T| = 2$. Then by a simple graph argument,

$$x(\delta(T)) = x(\overline{\gamma}(T \cup Q)) + x(\overline{\gamma}(T \cup Q')) = 2\overline{r} = 16$$

combining the above equation with the inequality (6.27) we have,

$$x(\overline{\gamma}(T \cup Q')) \geq 16 - 9 = 7 \tag{6.28}$$

Taking an arbitrary $Q \subseteq Y$, $|Q| = 5$ we have,

$$x(\delta(Q)) = x(\overline{\gamma}(T \cup Q)) + x(\overline{\gamma}(T' \cup Q)) = 5\overline{k} = 20$$

again by the biclique inequality (6.27) it follows,

$$x(\overline{\gamma}(T' \cup Q)) \geq 20 - 9 = 11 \tag{6.29}$$

Considering the complement of T as defined above,

$$x(\delta(T')) = x(\overline{\gamma}(T' \cup Q)) + x(\overline{\gamma}(T' \cup Q')) = 5\overline{r} = 40$$

applying the previously derived lower bound (6.29), it follows that,

$$x(\overline{\gamma}(T' \cup Q')) \leq 40 - 11 = 29 \tag{6.30}$$

The bounds with respect to the original incidence structure of a 2-(7,3,2) design can be obtained by supplementing variables in equations (6.27), (6.28), (6.29) and (6.30), respectively as follows,

$$10 - x(\gamma(T \cup Q)) \leq 9 \tag{6.31}$$

$$18 - x(\gamma(T \cup Q')) \geq 7 \tag{6.32}$$

$$25 - x(\gamma(T' \cup Q)) \geq 11 \tag{6.33}$$

$$45 - x(\gamma(T' \cup Q')) \leq 29 \tag{6.34}$$

Table 19 summarizes the bounds on the supplementing and the original structure for a design 2-(7,3,2) derived by the procedure explained in this section.

Table 19. Example of stronger classes from stable-set class for design 2-(7,3,2)

Class	$V(K)$	Class size	Vars in cut	Bound for $\|E(K)\|$
0	$K_{2,5}$	42042	10	lower: $l_0 = 1$
$\overline{0}'$	$K_{5,5}$	42042	25	upper: $u'_0 = 14$
$\overline{0}''$	$K_{2,9}$	42042	18	upper: $u''_0 = 11$
$\overline{0}'''$	$K_{5,9}$	42042	45	lower: $l'''_0 = 16$

The rest of the bounds can be derived using a similar procedure that in Subsection VI.1.3 where lower bounds were used. In this case, we will start with the lower bound on the class corresponding to the subgraph induced by $K_{t,b-(\overline{\lambda}+1)}$. For $G = (X \cup Y, E)$, let $K = G[T \cup Q]$ be an induced subgraph of G, with $|T| = t$ and $|Q| = b - (\overline{\lambda} + 1)$. The number of neighbors of T is given by 5.3. Consider another induced subgraph, $K' = G[T \cup Q']$ from the class we are analyzing. The smallest number of vertices in which Q and Q' intersect is the difference between the cardinality of Q' and the number of vertices outside the neighborhood $N(T)$. That is,

$$|Q \cap Q'| \geq b - (\overline{\lambda} + 1) - (b - |N(T)|)$$

Then, the number of edges with one end in T and the other end in Q is bounded below by this number, or by zero in case it is negative. For simplicity of notation, we call this lower bound β, that is,

$$x(E(K)) \geq \max\{0, |N(T)| - \overline{\lambda} - 1\} = \beta$$

We obtain the rest of the bounds as previously, by taking the complement, with respect to $G = (X \cup Y, E)$, of the subgraph $K = G[T \cup Q]$. The complements are three possible: one on $Y \setminus Q$, other on $X \setminus T$, and the last on both. The results are summarized in Table 20.

Table 20. Complement bounds for the stable-set classes

Class	$V(K)$	Class size	Bound for $\|E(K)\|$
$\overline{0}$	$K_{t,\overline{\lambda}+1}$	$\binom{v}{t}\binom{b}{\overline{\lambda}+1}$	upper: $tr - \beta$
$\overline{0}'$	$K_{v-t,\overline{\lambda}+1}$	$\binom{v}{t}\binom{b}{\overline{\lambda}+1}$	lower: $k(\overline{\lambda}+1) - tr + \beta$
$\overline{0}''$	$K_{t,b-(\overline{\lambda}+1)}$	$\binom{v}{t}\binom{b}{\overline{\lambda}+1}$	lower: $\max\{0, \|N(T)\| - \overline{\lambda} - 1\} = \beta$
$\overline{0}'''$	$K_{v-t,b-(\overline{\lambda}+1)}$	$\binom{v}{t}\binom{b}{\overline{\lambda}+1}$	upper: $k(b - \overline{\lambda} - 1) - \beta$

CHAPTER VII

STRONGER VALID INEQUALITIES FOR GDP FROM SUBSTRUCTURES

While working on the derivation of stronger classes of valid inequalities for GDP with the techniques shown in the previous chapter, we asked ourselves the question if applying the biclique inequalities of GDP to substructures of a t-design will provide a way to obtain other strong classes of valid inequalities. In this chapter we derive classes of valid inequalities for a certain type of substructures of a t-design. We will show that the bound for these inequalities are exact, and that, for a specific substructure, they are also implied equalities for the P_{star} polyhedron of GDP. These substructure results are valid for any s-design with $s < t$ but strongest for the case of 1-design substructures, and useful because 1-designs will be embedded in the incidence substructure of a t-design. In Section VII.1 we utilize the so called *derived* design substructure, and in Section VII.2 we use the *residual* design substructure. In section VII.3, we explore the idea of generating valid inequalities on the neighbors of blocks, instead of the neighbors of points. The concept of *dual* incidence structure is utilized for this purpose. Finally, in Section VIII.3 we give a pure cutting-plane algorithm that uses substructure cuts, along with an example of its implementation.

VII.1. Valid inequalities from derived design

It is a fact that the properties of a t-design are maintained no matter that the rows or columns are permuted in the corresponding point-block incidence matrix, therefore we claim that either a single point or a single block (or both) can be fixed a priori in the construction of the design. By fixing a point, we can obtain a *derived design*, which is defined as follows (see [37]).

Definition Given a collection $\mathcal{B}$ of k-element subsets of a v-element set X and a point $x \in X$, the *derivation* of $\mathcal{B}$ with respect to x is defined as,

$$\mathrm{DER}_x(\mathcal{B}) = \{B \setminus \{x\} : x \in B \in \mathcal{B}\} \tag{7.1}$$

The following Theorem from Kreher [37] relates a t-design and a corresponding derived design.

Theorem 25 (see [37]) *If there exists a t-(v, k, λ) design $(X, \mathcal{B})$ and $x \in X$ is any point, then $(X \setminus \{x\}, \mathrm{DER}_x(\mathcal{B}))$ is a $(t-1)$-$(v-1, k-1, \lambda)$ design.*

An example of a derived design of the design 2-(10,4,2) $b = 15$, $r = 6$ with respect to the first point is given in Figure 9. The framed portion of the point-block incidence matrix is the derived design 1-(9,3,2) on 6 blocks.

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	1	1	1	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	1	1	1	0	0
1	0	0	1	0	0	0	1	0	0	1	0	0	1	1
0	1	1	0	0	0	0	0	1	0	0	1	0	1	1
0	1	0	0	1	0	0	0	0	1	1	0	1	1	0
0	0	1	0	0	1	0	1	0	1	0	0	1	0	1
0	0	0	1	1	0	1	0	1	0	0	0	1	0	1
0	0	0	1	0	1	1	0	0	1	0	1	0	1	0
0	0	0	0	1	1	0	1	1	0	1	1	0	0	0

Fig. 9. Example of derived design of 2-(10,4,2) with respect to the first point.

Applying our GDP formulation biclique inequalities to the derived design, we obtain the following,

$$x(\gamma(V(K_{t-1,\lambda+1}))) \leq (t-1)(\lambda+1)-1 \tag{7.2}$$

We can obtain a derived design of a derived design, because there is a theorem for t-designs that states that a t-design is a s-design for $0 \leq s \leq t$ (see Theorem 1). Therefore (7.2) can be applied recursively for all the derived designs of a t-design. Since we are using the biclique inequalities in GDP for the derivation, the smallest meaningful inequality would be the one on a single point. The inequalities for derived designs are then,

$$x(\gamma(V(K_{t-s,\lambda+1}))) \leq (t-s)(\lambda+1)-1, \quad s = 1, 2, \ldots, t-1 \qquad \text{(Derived classes)}$$

For the specific case of $s = t-1$, the above inequalities (Derived classes) are stated as follows:

$$x(E(K_{1,\lambda+1})) \leq \lambda \tag{7.3}$$

One of the main results in this chapter is the derivation of a stronger version of the inequalities (7.3). We will show that any star inequality induced by a star graph of fewer vertices than $V(K_{1,b})$ or $V(K_{v,1})$ for a t-(v,k,λ) design on b blocks, is redundant to the star inequalites in the formulation GDP. That implies that inequality (7.3) can be extended or *lifted* to the block-star inequalities in GDP for the particular derived 1-design. The same applies for the case of the point-star inequalities, as we will show when using the dual of a design. The result if the following,

Proposition 26 *For a t-(v,k,λ) design with b blocks, a star inequality for bicliques of the form $x(E(K_{1,m})) \leq u$ for $m < b$, is redundant with the star inequalities $x(E(K_{1,b})) \leq u$.*

Proof. Suppose, by way of contradiction, that $x(E(K_{1,b})) \leq u$ but $x(E(K_{1,m})) > u$. Partition the block set of $K_{1,b}$ into two sets, one of m elements (corresponding to $K_{1,m}$) and the other of $(b-m)$ elements. Then,

$$u \geq x(E(K_{1,b})) = x(E(K_{1,m})) + x(E(K_{1,b-m})) > u + x(E(K_{1,b-m})$$

a contradiction because $x(E(K_{1,b-m}))$ cannot be negative. □

Note that, given a t-(v,k,λ) design, the number of blocks in every derived $(t-s)$-design from it is λ_s for $s = 0, 1, \ldots, t$ (see equation (2.1) for the definition of λ_s). In particular, for $s = t-1$, the number of blocks in the derived 1-design is $\lambda_{(t-1)}$. By Proposition 26, the lifted version of the derived 1-design inequality (7.3) is therefore,

$$\boxed{x(E(K_{1,\lambda_{(t-1)}})) \leq \lambda} \qquad \text{(Lifted derived star class)}$$

The main result about inequalities (Lifted derived star class) for t-designs is that they are implied equations for GDP, that is, that they are satisfied as equality by every feasible solution. Our result is stated as follows,

Proposition 27 *The derived 1-design inequalities (Lifted derived star class) are implied equations with respect to GDP for a t-design.*

Proof.

Consider the supplementary structure as defined in Section VI.2. With respect to the original t-(v,k,λ) design, the supplementary structure of the derived 1-design 1-$(v-t+1, k-t+1, \lambda)$ will be a 1-$(v-t+1, v-k, \overline{\lambda'})$, where $\overline{\lambda'}$ is obtained from the definition given in (6.24) and simplified as follows,

$$\overline{\lambda'} = \frac{\lambda(v-k)}{k-t+1} \tag{7.4}$$

The number of blocks of this derived 1-design is the same for the supplement of it, and is $\lambda_{(t-1)}$, as stated earlier. An expression for this number is obtained from (2.1) and simplified as follows:

$$\lambda_{(t-1)} = \frac{\lambda(v-t+1)}{k-t+1} \tag{7.5}$$

Applying the biclique inequalities of GDP to the supplement of the 1-design we obtain,

$$x(E(\overline{K}_{1,\overline{\lambda'}+1})) \leq \overline{\lambda'} \tag{7.6}$$

Again, by proposition 26, the above inequality can be lifted to the entire set of blocks of the supplementary 1-design as follows,

$$x(E(\overline{K}_{1,\lambda_{(t-1)}})) \leq \overline{\lambda'} \tag{7.7}$$

By complementing the binary variables in (7.8), we obtain and expression for the original 1-design,

$$(1)(\lambda_{(t-1)}) - x(E(K_{1,\lambda_{(t-1)}})) \leq \overline{\lambda'} \tag{7.8}$$

rearranging we have,

$$x(E(K_{1,\lambda_{(t-1)}})) \geq \lambda_{(t-1)} - \overline{\lambda'} \tag{7.9}$$

Using equations (7.4) and (7.5), the right-hand side of the above inequality is

$$\lambda_{(t-1)} - \overline{\lambda'} = \frac{\lambda(v-t+1)}{k-t+1} - \frac{\lambda(v-k)}{k-t+1} = \lambda \tag{7.10}$$

Then, equation (7.9) can be written as,

$$x(E(K_{1,\lambda_{(t-1)}})) \geq \lambda \tag{7.11}$$

Therefore, by equations (7.11) and (Lifted derived star class) it follows that the 1-derived design inequalities are implied equations for GDP for a t-design. □

In the example given in Figure 9, this result means that the number of ones per row in the derived design 1-(9,3,2) (and any other derived design with respect to other points) is constant and equal to 2, for all rows.

VII.2. Valid inequalities from residual design

In this section we derive other strong valid inequalities from substructures very similar to the ones proved in the previous Section VII.1. We start with the definition, found in Kreher [37], of the substructures of a t-design that we study here.

Definition Given a collection $\mathcal{B}$ of k-element subsets of a v-element set X and a point $x \in X$, the *residual* of $\mathcal{B}$ with respect to x is defined as,

$$\mathrm{RES}_x(\mathcal{B}) = \{B \in \mathcal{B} : x \notin B\} \tag{7.12}$$

The following Theorem (see [37]) relates a t-design and a corresponding *residual design*.

Theorem 28 *If there exists a t-(v, k, λ) design $(X, \mathcal{B})$ and $x \in X$ is any point, then $(X \setminus \{x\}, \mathrm{RES}_x(\mathcal{B}))$ is a $(t-1)$-$(v-1, k, \frac{\lambda(v-k)}{k-t+1})$ design.*

Figure 10 shows the point-block incidence matrix of a 2-(10,4,2) design on 15 blocks. The residual design with respect to the first point is framed in the matrix. The residual design is a 1-(9,4,4) design on 9 blocks.

As pointed out in the previous section, in the same way we can obtain the derived of a derived design, the residual of a residual design can also be obtained. We derive a formula for the index parameter (i.e. lambda) for a residual $(t-s)$-design, $s = 0, 1, 2, \ldots, t$ with respect to the original t-(v, k, λ) design. Since the index parameter is a function of the respective parameters of every residual de-

$$
\begin{array}{cccccc|ccccccccc|}
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cline{7-15}
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
\cline{7-15}
\end{array}
$$

Fig. 10. Example of residual design of 2-(10,4,2) with respect to the first point.

sign, the formula needs to be obtained recursively. The closed-form expression we obtained is as follows:

$$\lambda^{[t-s]} = \frac{\lambda\binom{v-t}{k-t+s}}{\binom{v-t}{k-t}}, \quad s = 0, 1, \ldots, t \tag{7.13}$$

The biclique inequalities of GDP are relevant for any t-design with $t > 0$, then by applying those biclique inequalities to all the possible residual $(t-s)$-designs with $s = 1, 2, \ldots, t-1$, we obtain the residual design classes of valid inequalities as follows,

$$x(\gamma(V(K_{t-s,\lambda^{[t-s]}+1}))) \leq (t-s)(\lambda^{[t-s]}+1)-1, \quad s = 1, 2, \ldots, t-1 \quad \text{(Residual classes)}$$

Considering the case $s = t-1$, the residual is a 1-($v-t+1, k, \lambda^{[1]}$)) design. Then the above (Residual classes) reduces to an inequality for a residual 1-design,

$$x(E(K_{1,\lambda^{[1]}+1})) \leq \lambda^{[1]} \tag{7.14}$$

In general, the number of blocks of a residual $(t-s)$-design with respect to the original t-(v,k,λ) design, can be obtained from (2.1) and (7.13), and is equal to:

$$b^{[t-s]} = \frac{\lambda^{[t-s]}\binom{v-s}{t-s}}{\binom{k}{t-s}} = \frac{\lambda\binom{v-s}{k}}{\binom{v-t}{v-k}}, \quad s = 0, 1, \ldots, t \tag{7.15}$$

In particular, when $s = t-1$, the number of blocks of a residual 1-design, $b^{[1]}$, with respect to the original t-(v,k,λ) design is equal to:

$$b^{[1]} = \frac{\lambda\binom{v-t+1}{k}}{\binom{v-t}{v-k}} \tag{7.16}$$

By Proposition 26, inequality (7.14) can be lifted to obtain,

$$\boxed{x(E(K_{1,b^{[1]}})) \leq \lambda^{[1]}} \qquad \text{(Lifted residual star class)}$$

Note that the supplementing structure of the residual design with respect to the original t-(v,k,λ) design, is a $(t-1)$-$(v-1, v-k-1, \frac{\lambda\binom{v-k}{t}}{\binom{k}{t}})$ design. In particular, for the residual 1-design 1-$(v-t+1, k, \lambda^{[1]})$, the supplementing structure is a 1-$(v-t+1, v-t+1-k, \overline{\lambda^{[1]}})$, where

$$\overline{\lambda^{[1]}} = \frac{\lambda^{[1]}(v-t-k+1)}{k} \tag{7.17}$$

obtained from the definition of $\overline{\lambda}$ in a supplementary structure (see equation (6.24)). We can show that the residual 1-design inequalities (Lifted residual star class) are implied equalities for GDP following the same procedure used for the derived 1-design inequalities (i.e. the proof of Proposition 27). The main result is the following,

Proposition 29 *The residual 1-design inequalities (Lifted residual star class) are implied equations with respect to GDP for a t-design.*

Proof. Applying (Lifted residual star class) to the supplement of the residual 1-design,

$$x(E(\overline{K}_{1,b^{[1]}})) \leq \overline{\lambda^{[1]}}$$

By complementing binary variables in the above inequality,

$$(1)(b^{[1]}) - x(E(K_{1,b^{[1]}})) \leq \overline{\lambda^{[1]}}$$

and rearranging

$$x(E(K_{1,b^{[1]}})) \geq b^{[1]} - \overline{\lambda^{[1]}}$$

The right-hand side of the above inequality can be simplified using (7.16) and (7.17) to:

$$b^{[1]} - \overline{\lambda^{[1]}} = \frac{\lambda^{[1]}(v-t+1)}{k} - \frac{\lambda^{[1]}(v-t+1-k)}{k} = \lambda^{[1]}$$

Then,

$$x(E(K_{1,b^{[1]}})) \geq \lambda^{[1]} \tag{7.18}$$

The result follows by the above equation (7.18) and equation (Lifted residual star class). □

VII.3. Valid inequalities from dual design

In the two previous sections of this chapter, we derived substructure inequalities for neighborhoods of points. In this section, we will introduce the concept of dual design to define other classes of valid inequalities for substructures utilizing neighborhoods of blocks. The bounds obtained for these inequalities are exact, and they are implied equations for GDP, as we will show. We start with a definition from design theory (see [55]).

Definition The *dual design* of a t-design is obtained by interchanging the roles of points and blocks.

If general, if D is a t-design with point-block incidence matrix X, then the transpose matrix $X^\top$ defines de dual design $\tilde{D}$ of D.

Theorem 30 (see [55]) *The dual of a t-(v, k, λ) design with b blocks and point replication r, is a 1-(b, r, k) design, provided that $t \geq 1$.*

The above theorem may seem not very strong, since it only tells us that a t-design is a b-factor. Anyway, we will use this definition to create classes of valid inequalities for star subgraphs with points as leaves.

As pointed out in Section VII.1, given a t-(v, k, λ) design, any derived $(t-s)$-$(v-s, k-s, \lambda)$ design for $s = 0, 1, \ldots, t$ has λ_s blocks and λ_{s+1} point replication parameter (see page 7 for the definition of λ_s). Then, by the above theorem, the dual of such derived $(t-s)$-design is a 1-$(\tilde{v}, \tilde{k}, \tilde{\lambda})$ design where $\tilde{v} = \lambda_s$, $\tilde{k} = \lambda_{s+1}$, and $\tilde{\lambda} = k - s$. That is, the dual of a derived $(t-s)$-design is a 1-$(\lambda_s, \lambda_{s+1}, k-s)$ design on $v - s$ blocks.

We apply the original biclique inequalities of GDP now for the dual of a derived design. In this case, the points are blocks and the blocks are points, so to keep the original point-block incidence matrix, the subgraphs are now induced by $V(K_{\lambda+1,t})$. Then for the 1-$(\lambda_s, \lambda_{s+1}, k-s)$ design,

$$x(E(K_{k-s+1,1})) \leq k - s, \quad s = 1, 2, \ldots, t-1 \tag{7.19}$$

The above inequality can be lifted by Proposition 26 to the entire set of blocks of the dual-derived design (that is, points in the derived design), to have $v - s$

instead of of $k - s + 1$ dual-derived blocks. The lifted version of (7.19) for the dual of a derived $(t - s)$-design is then,

$$\boxed{x(E(K_{v-s,1})) \leq k - s, \quad s = 1, 2, \ldots, t-1} \qquad \text{(Lifted dual-derived star class)}$$

To show (Lifted dual-derived star class) are implied equations for GDP, we need to use again the definition of supplementary structure from Section VI.2. Given a 1-(v, k, λ) design, the supplement design will be a 1-$(v, v - k, \overline{\lambda}_1)$ where the index parameter is obtained from (6.24) with $s = t = 1$ as follows,

$$\overline{\lambda}_1 = \frac{\lambda\binom{v-1}{k}}{\binom{v-1}{k-1}} = \frac{\lambda(v-k)}{k} \tag{7.20}$$

Our result is,

Proposition 31 *The dual-derived 1-design inequalities (Lifted dual-derived star class) are implied equations with respect to GDP for a t-design.*

Proof. Given the dual design 1-$(\lambda_s, \lambda_{s+1}, k - s)$, the supplement of it is therefore a 1-$(\lambda_s, \lambda_{s+1} - \lambda_s, v - k)$ design, where the index parameter was computed using (7.20) and simplified using the definition of λ_s (2.1) as,

$$\frac{(k-s)(\lambda_s - \lambda_{s+1})}{\lambda_{s+1}} = \frac{(k-s)\lambda_{s+1}(v-k)/(k-s)}{\lambda_{s+1}} = v - k$$

Applying (Lifted dual-derived star class) to this supplementary design we get,

$$x(E(\overline{K}_{v-s,1})) \leq v - k \tag{7.21}$$

Complementing the binary variables,

$$(v-s)(1) - x(E(K_{v-s,1})) \leq v - k \tag{7.22}$$

and rearranging we obtain,

$$x(E(K_{v-s,1})) \geq v - s - (v - k) = k - s \tag{7.23}$$

By equations (Lifted dual-derived star class) and (7.23), we can conclude that the inequalities in this class are implied equalities for GDP. □

We give also dual inequalities for the residual design. As pointed out in Section VII.2, given a t-(v, k, λ) design, a residual design is a $(t - s)$-$(v - s, k, \lambda^{[t-s]})$ design for $s = 0, 1, \ldots, t$, where the index parameter is computed as defined in equation (7.13). The number of blocks of every residual design is given in (7.15). The point replication parameter $r^{[t-s]}$ for every residual design can be derived in the same way as (7.15) to give,

$$r^{[t-s]} = \frac{\lambda^{[t-s]}\binom{v-s-1}{t-s-1}}{\binom{k-1}{t-s-1}} = \frac{\lambda\binom{v-s-1}{k-1}}{\binom{v-t}{v-k}}, \quad s = 0, 1, \ldots, t \tag{7.24}$$

Therefore, by Theorem 30, the dual of every such residual $(t - s)$-designs for $s = 0, 1, \ldots, t$ is a 1-$(b^{[t-s]}, r^{[t-s]}, k)$ on $v - s$ blocks. We apply the original biclique inequalities to these dual 1-designs. Again, the role of points and blocks is interchanged, so to be consistent with the original order of the point-block incidence matrix of a design, the induced subgraphs will be of the form $V(K_{\lambda+1,1})$. Then for a 1-$(b^{[t-s]}, r^{[t-s]}, k)$ design, the corresponding biclique inequalities are,

$$x(E(K_{k+1,1})) \leq k$$

which can be lifted by Proposition 26 to its entire set of $(v - s)$ dual-residual blocks to the following,

$$\boxed{x(E(K_{v-s,1})) \leq k, \quad s = 1, 2, \ldots, t-1} \qquad \text{(Lifted dual-residual star class)}$$

This class are also implied equation, as shown in the next result.

Proposition 32 *The dual-residual 1-design inequalities (Lifted dual-residual star class) are implied equations with respect to GDP for a t-design.*

Proof. The supplement of the dual design 1-($b^{[t-s]}, r^{[t-s]}, k$) is a 1-($b^{[t-s]}, b^{[t-s]} - r^{[t-s]}, v - k - s$) design. The index parameter was computed using (7.20) and simplified using the definition of λ_s (2.1) as,

$$\frac{k(b^{[t-s]} - r^{[t-s]})}{r^{[t-s]}} = \frac{kr^{[t-s]}(v-k-s)/k}{r^{[t-s]}} = v - k - s$$

Applying (Lifted dual-residual star class) to this supplementary design we get,

$$x(E(\overline{K}_{v-s,1})) \leq v - k - s \tag{7.25}$$

Complementing the binary variables,

$$(v - s)(1) - x(E(K_{v-s,1})) \leq v - k - s \tag{7.26}$$

and rearranging we obtain,

$$x(E(K_{v-s,1})) \geq v - s - (v - k - s) = k \tag{7.27}$$

The result follows by equations (Lifted dual-residual star class) and (7.27). □

VII.4. A cutting-plane algorithm using cuts from substructures

In view of the strength of the results of the previous sections, we devised an algorithm that uses substructure cuts and adds them iteratively to obtain the solution at the root node of the search tree. That is, a pure cutting-plane algorithm which is presented in Figure 11. The algorithm receives as input the parameters for a t-design, an initial arbitrary partition, and a list of subsets of points of size $(t-1)$ in transpositional order. It returns either a solution or failure.

Considering the point-block matrix representation of a t-designs, the subroutine in Figure 12 generates a partition of the columns of the matrix based on the information that gives a particular subset of rows of size $(t-1)$. This subset of rows tells which 1-design to consider, among the possible 2^{t-1} 1-design substructures, by converting its binary value to decimal. Once the corresponding 1-design is determined, the corresponding cuts are added. The procedure continues for the rest of the point subset of size $(t-1)$, until a design is found or the current LP solution is infeasible or non-integral.

We define the following variables and data for the algorithm in Figure 11:

- point set: $V = \{1, 2, \ldots v\}$
- number of subsets in block-partition: $p = 2^{t-1}$
- block-partition: $\Pi = \{\pi_0, \pi_1, \ldots, \pi_{p-1}\}$
- number of point-sets of size $(t-1)$: $n = \binom{v}{t-1}$
- point-sets of size $(t-1)$ in transpositional order: $\mathcal{L} = \{L^0, L^1, \ldots, L^{n-1}\}$
- point-block incidence matrix: D

The success of this cutting-plane algorithm depends on the LP solution being integral, as it should be provided that the partitions given are correct. But knowing these partitions would be equivalent to, and therefore as difficult as, knowing the complete polyhedral description of the problem of finding a t-design, since it could be retrieved form the partitions.

Input: $(t, v, k, \lambda), \Pi^0, \mathcal{L}$
substructures: $p \leftarrow 2^{t-1}$
subsets: $n \leftarrow \binom{v}{t-1}$
iteration: $i \leftarrow 0$
current partition: $\Pi \leftarrow \Pi^0$ {initial partition is arbitrary}
current subset of points: $L \leftarrow L^0 \in \mathcal{L}$
while $i < n$ **do**
  for every block-subset $\pi \in \Pi^i$ **do**
    add inequalities on points $V \setminus L$.
  end for
  Solve LP.
  if LP is non-integral or infeasible **then**
    EXIT. {failure}
  else
    store solution in point-block incidence matrix D.
  end if
  if solution if a t-(v, k, λ) design **then**
    print D.
    EXIT. {success}
  else
    $\Pi \leftarrow$ UPDATEPARTITION(D, L)
    $L \leftarrow L^{i+1}$
    $i \leftarrow i + 1$
  end if
end while

Fig. 11. Pure cutting-plane algorithm with substructure cuts

Input: (D, L).
Output: Updated partition Π^{new}.
$\Pi^{new} \leftarrow \emptyset$.
for all block j of D **do**
  $d \leftarrow$ decimal value of j on rows L.
  add block j to subset $\pi_d \in \Pi^{new}$.
end for

Fig. 12. Update partition subroutine

VII.4.1. Example for design 3-(8,4,2)

We present an example on the implementation of the cutting-plane algorithm in Figure 11, for the specific case of a design 3-(8,4,2) on 28 blocks and point replication 14. We choose this instance to exemplify that this cutting-plane procedure goes beyond the most studied instances with $\lambda = 1$ (see [46, 39, 38]), and beyond block designs $(t = 2)$ that were approached by a cutting-plane algorithm before (see [47]).

The total 1-design substructures in this example are $2^2 = 4$. Table 21 gives them, their decimal and binary representation, and the right-hand-side value for the cut. This value is obtained from equations (Lifted derived star class) or (Lifted residual star class).

Table 21. Substructures for a 3-(8,4,2) design

Decimal	binary	substructure	1-design	rhs
0	00	res-res	1-(6,4,4)	4
1	01	res-der	1-(6,3,4)	4
2	10	der-res	1-(6,3,4)	4
3	11	der-der	1-(6,2,2)	2

The base formulation are the star inequalities in GDP. There are 28 point-star inequalities, and 8 block-star inequalities. In this example, we used equalities since the inequalities are implied, as previously show. The total number of variables in the optimization model is 224. The linear programming LP solution is obtained with OSL's simplex solver [32]. The point-subsets in transpositional order are:

$$\mathcal{L} = \{\{0,1\},\{1,2\},\{0,2\},\{2,3\},\{1,3\},\{0,3\},\ldots,\{0,7\}\}$$

Base formulation:

Star inequalities, points-as-leaves:

$$x_{0,0} + x_{1,0} + x_{2,0} + x_{3,0} + x_{4,0} + x_{5,0} + x_{6,0} + x_{7,0} = 4$$

$$x_{0,1} + x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} + x_{6,1} + x_{7,1} = 4$$

$$x_{0,2} + x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} + x_{6,2} + x_{7,2} = 4$$

$$\vdots$$

$$x_{0,27} + x_{1,27} + x_{2,27} + x_{3,27} + x_{4,27} + x_{5,27} + x_{6,27} + x_{7,27} = 4$$

Star inequalities, blocks-as-leaves:

$$x_{0,0} + x_{0,1} + x_{0,2} + \cdots + x_{0,26} + x_{0,27} = 14$$

$$x_{1,0} + x_{1,1} + x_{1,2} + \cdots + x_{1,26} + x_{1,27} = 14$$

$$x_{2,0} + x_{2,1} + x_{2,2} + \cdots + x_{2,26} + x_{2,27} = 14$$

$$x_{3,0} + x_{3,1} + x_{3,2} + \cdots + x_{3,26} + x_{3,27} = 14$$

$$x_{4,0} + x_{4,1} + x_{4,2} + \cdots + x_{4,26} + x_{4,27} = 14$$

$$x_{5,0} + x_{5,1} + x_{5,2} + \cdots + x_{5,26} + x_{5,27} = 14$$

$$x_{6,0} + x_{6,1} + x_{6,2} + \cdots + x_{6,26} + x_{6,27} = 14$$

$$x_{7,0} + x_{7,1} + x_{7,2} + \cdots + x_{7,26} + x_{7,27} = 14$$

Iteration 0:

Partition $\Pi^0 = \{\pi_0, \pi_1, \pi_2, \pi_3\}$:

$$\pi_0 = \{22, 23, 24, 25, 26, 27\}$$

$$\pi_1 = \{14, 15, 16, 17, 18, 19, 20, 21\}$$

$$\pi_2 = \{6, 7, 8, 9, 10, 11, 12, 13\}$$

$$\pi_3 = \{0, 1, 2, 3, 4, 5\}$$

Point combination: $L^0 = \{0, 1\}$ Added cuts:

$$x_{2,22} + x_{2,23} + x_{2,24} + x_{2,25} + x_{2,26} + x_{2,27} = 4$$

$$x_{3,22} + x_{3,23} + x_{3,24} + x_{3,25} + x_{3,26} + x_{3,27} = 4$$

$$x_{4,22} + x_{4,23} + x_{4,24} + x_{4,25} + x_{4,26} + x_{4,27} = 4$$

$$x_{5,22} + x_{5,23} + x_{5,24} + x_{5,25} + x_{5,26} + x_{5,27} = 4$$

$$x_{6,22} + x_{6,23} + x_{6,24} + x_{6,25} + x_{6,26} + x_{6,27} = 4$$

$$x_{7,22} + x_{7,23} + x_{7,24} + x_{7,25} + x_{7,26} + x_{7,27} = 4$$

$$x_{2,14} + x_{2,15} + x_{2,16} + x_{2,17} + x_{2,18} + x_{2,19} + x_{2,20} + x_{2,21} = 4$$

$$x_{3,14} + x_{3,15} + x_{3,16} + x_{3,17} + x_{3,18} + x_{3,19} + x_{3,20} + x_{3,21} = 4$$

$$x_{4,14} + x_{4,15} + x_{4,16} + x_{4,17} + x_{4,18} + x_{4,19} + x_{4,20} + x_{4,21} = 4$$

$$x_{5,14} + x_{5,15} + x_{5,16} + x_{5,17} + x_{5,18} + x_{5,19} + x_{5,20} + x_{5,21} = 4$$

$$x_{6,14} + x_{6,15} + x_{6,16} + x_{6,17} + x_{6,18} + x_{6,19} + x_{6,20} + x_{6,21} = 4$$

$$x_{7,14} + x_{7,15} + x_{7,16} + x_{7,17} + x_{7,18} + x_{7,19} + x_{7,20} + x_{7,21} = 4$$

$$x_{2,6} + x_{2,7} + x_{2,8} + x_{2,9} + x_{2,10} + x_{2,11} + x_{2,12} + x_{2,13} = 4$$

$$x_{3,6} + x_{3,7} + x_{3,8} + x_{3,9} + x_{3,10} + x_{3,11} + x_{3,12} + x_{3,13} = 4$$

$$x_{4,6} + x_{4,7} + x_{4,8} + x_{4,9} + x_{4,10} + x_{4,11} + x_{4,12} + x_{4,13} = 4$$

$$x_{5,6} + x_{5,7} + x_{5,8} + x_{5,9} + x_{5,10} + x_{5,11} + x_{5,12} + x_{5,13} = 4$$

$$x_{6,6} + x_{6,7} + x_{6,8} + x_{6,9} + x_{6,10} + x_{6,11} + x_{6,12} + x_{6,13} = 4$$

$$x_{7,6} + x_{7,7} + x_{7,8} + x_{7,9} + x_{7,10} + x_{7,11} + x_{7,12} + x_{7,13} = 4$$

$$x_{2,0} + x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} = 2$$

$$x_{3,0} + x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} + x_{3,5} = 2$$

$$x_{4,0} + x_{4,1} + x_{4,2} + x_{4,3} + x_{4,4} + x_{4,5} = 2$$

$$x_{5,0} + x_{5,1} + x_{5,2} + x_{5,3} + x_{5,4} + x_{5,5} = 2$$

$$x_{6,0} + x_{6,1} + x_{6,2} + x_{6,3} + x_{6,4} + x_{6,5} = 2$$

$$x_{7,0} + x_{7,1} + x_{7,2} + x_{7,3} + x_{7,4} + x_{7,5} = 2$$

LP solution:

$$D^0 = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . \\
1 & . & 1 & . & . & . & . & 1 & 1 & 1 & 1 & . & . & . & 1 & 1 & . & 1 & . & 1 & . & . & 1 & . & 1 & 1 & 1 & . \\
1 & 1 & . & . & . & . & 1 & 1 & 1 & . & 1 & . & . & . & . & . & 1 & 1 & 1 & 1 & . & . & 1 & 1 & 1 & . & 1 & . \\
. & 1 & . & 1 & . & . & 1 & 1 & . & 1 & . & 1 & . & . & . & . & 1 & . & 1 & . & 1 & 1 & 1 & 1 & . & 1 & . & 1 \\
. & . & 1 & . & 1 & . & 1 & . & . & . & . & 1 & 1 & 1 & . & 1 & . & 1 & 1 & . & 1 & . & . & 1 & 1 & . & 1 & 1 \\
. & . & . & 1 & . & 1 & . & . & . & 1 & . & 1 & 1 & 1 & 1 & . & 1 & . & . & . & 1 & 1 & . & 1 & 1 & 1 & . & 1 \\
. & . & . & . & 1 & 1 & . & . & 1 & . & 1 & . & 1 & 1 & 1 & 1 & . & . & . & 1 & . & 1 & 1 & . & . & 1 & 1 & 1
\end{bmatrix}$$

Rows = 60, Objective = 112, Bicliques remaining = 36.

Iteration 1:

Partition $\Pi^1 = \{\pi_0, \pi_1, \pi_2, \pi_3\}$:

$$\pi_0 = \{6, 11, 12, 13, 23, 27\}$$

$$\pi_1 = \{7, 8, 9, 10, 22, 24, 25, 26\}$$

$$\pi_2 = \{1, 3, 4, 5, 16, 18, 20, 21\}$$

$$\pi_3 = \{0, 2, 14, 15, 17, 19\}$$

Point combination: $L^1 = \{1, 2\}$

Added cuts:

$$x_{0,6} + x_{0,11} + x_{0,12} + x_{0,13} + x_{0,23} + x_{0,27} = 4$$

$$x_{3,6} + x_{3,11} + x_{3,12} + x_{3,13} + x_{3,23} + x_{3,27} = 4$$

$$x_{4,6} + x_{4,11} + x_{4,12} + x_{4,13} + x_{4,23} + x_{4,27} = 4$$

$$x_{5,6} + x_{5,11} + x_{5,12} + x_{5,13} + x_{5,23} + x_{5,27} = 4$$

$$x_{6,6} + x_{6,11} + x_{6,12} + x_{6,13} + x_{6,23} + x_{6,27} = 4$$

$$x_{7,6} + x_{7,11} + x_{7,12} + x_{7,13} + x_{7,23} + x_{7,27} = 4$$

$$x_{0,7} + x_{0,8} + x_{0,9} + x_{0,10} + x_{0,22} + x_{0,24} + x_{0,25} + x_{0,26} = 4$$

$$x_{3,7} + x_{3,8} + x_{3,9} + x_{3,10} + x_{3,22} + x_{3,24} + x_{3,25} + x_{3,26} = 4$$

$$x_{4,7} + x_{4,8} + x_{4,9} + x_{4,10} + x_{4,22} + x_{4,24} + x_{4,25} + x_{4,26} = 4$$

$$x_{5,7} + x_{5,8} + x_{5,9} + x_{5,10} + x_{5,22} + x_{5,24} + x_{5,25} + x_{5,26} = 4$$

$$x_{6,7} + x_{6,8} + x_{6,9} + x_{6,10} + x_{6,22} + x_{6,24} + x_{6,25} + x_{6,26} = 4$$

$$x_{7,7} + x_{7,8} + x_{7,9} + x_{7,10} + x_{7,22} + x_{7,24} + x_{7,25} + x_{7,26} = 4$$

$$x_{0,1} + x_{0,3} + x_{0,4} + x_{0,5} + x_{0,16} + x_{0,18} + x_{0,20} + x_{0,21} = 4$$

$$x_{3,1} + x_{3,3} + x_{3,4} + x_{3,5} + x_{3,16} + x_{3,18} + x_{3,20} + x_{3,21} = 4$$

$$x_{4,1} + x_{4,3} + x_{4,4} + x_{4,5} + x_{4,16} + x_{4,18} + x_{4,20} + x_{4,21} = 4$$

$$x_{5,1} + x_{5,3} + x_{5,4} + x_{5,5} + x_{5,16} + x_{5,18} + x_{5,20} + x_{5,21} = 4$$

$$x_{6,1} + x_{6,3} + x_{6,4} + x_{6,5} + x_{6,16} + x_{6,18} + x_{6,20} + x_{6,21} = 4$$

$$x_{7,1} + x_{7,3} + x_{7,4} + x_{7,5} + x_{7,16} + x_{7,18} + x_{7,20} + x_{7,21} = 4$$

$$x_{0,0} + x_{0,2} + x_{0,14} + x_{0,15} + x_{0,17} + x_{0,19} = 2$$

$$x_{3,0} + x_{3,2} + x_{3,14} + x_{3,15} + x_{3,17} + x_{3,19} = 2$$

$$x_{4,0} + x_{4,2} + x_{4,14} + x_{4,15} + x_{4,17} + x_{4,19} = 2$$

$$x_{5,0} + x_{5,2} + x_{5,14} + x_{5,15} + x_{5,17} + x_{5,19} = 2$$

$$x_{6,0} + x_{6,2} + x_{6,14} + x_{6,15} + x_{6,17} + x_{6,19} = 2$$

$$x_{7,0} + x_{7,2} + x_{7,14} + x_{7,15} + x_{7,17} + x_{7,19} = 2$$

LP solution:

$$D^1 = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . \\
1 & . & 1 & . & . & . & . & 1 & 1 & 1 & 1 & . & . & . & 1 & 1 & . & 1 & . & 1 & . & . & 1 & . & 1 & 1 & 1 & . \\
1 & 1 & . & . & . & . & 1 & . & . & . & 1 & 1 & 1 & . & . & 1 & 1 & . & 1 & . & 1 & . & 1 & 1 & . & 1 & 1 & . \\
. & 1 & . & 1 & . & . & 1 & 1 & . & 1 & . & 1 & . & . & 1 & . & 1 & . & . & 1 & 1 & . & 1 & 1 & . & 1 & . & 1 \\
. & . & 1 & . & 1 & . & . & 1 & 1 & . & . & 1 & . & 1 & . & . & . & 1 & 1 & . & 1 & 1 & . & 1 & 1 & . & 1 & 1 \\
. & . & . & 1 & . & 1 & . & . & 1 & 1 & . & . & 1 & 1 & 1 & . & . & 1 & 1 & . & . & 1 & . & 1 & 1 & 1 & . & 1 \\
. & . & . & . & 1 & 1 & 1 & . & . & . & 1 & . & 1 & 1 & . & 1 & 1 & . & . & 1 & . & 1 & 1 & . & 1 & . & 1 & 1
\end{bmatrix}$$

Rows = 84, Objective = 112, Bicliques remaining = 26.

Iteration 2:

Partition $\Pi^2 = \{\pi_0, \pi_1, \pi_2, \pi_3\}$:

$$\pi_0 = \{16, 18, 20, 21, 23, 27\}$$

$$\pi_1 = \{14, 15, 17, 19, 22, 24, 25, 26\}$$

$$\pi_2 = \{1, 3, 4, 5, 6, 11, 12, 13\}$$

$$\pi_3 = \{0, 2, 7, 8, 9, 10\}$$

Point combination: $L^2 = \{0, 2\}$

Added cuts:

$$x_{1,16} + x_{1,18} + x_{1,20} + x_{1,21} + x_{1,23} + x_{1,27} = 4$$

$$x_{3,16} + x_{3,18} + x_{3,20} + x_{3,21} + x_{3,23} + x_{3,27} = 4$$

$$x_{4,16} + x_{4,18} + x_{4,20} + x_{4,21} + x_{4,23} + x_{4,27} = 4$$

$$x_{5,16} + x_{5,18} + x_{5,20} + x_{5,21} + x_{5,23} + x_{5,27} = 4$$

$$x_{6,16} + x_{6,18} + x_{6,20} + x_{6,21} + x_{6,23} + x_{6,27} = 4$$

$$x_{7,16} + x_{7,18} + x_{7,20} + x_{7,21} + x_{7,23} + x_{7,27} = 4$$

$$x_{1,14} + x_{1,15} + x_{1,17} + x_{1,19} + x_{1,22} + x_{1,24} + x_{1,25} + x_{1,26} = 4$$

$$x_{3,14} + x_{3,15} + x_{3,17} + x_{3,19} + x_{3,22} + x_{3,24} + x_{3,25} + x_{3,26} = 4$$

$$x_{4,14} + x_{4,15} + x_{4,17} + x_{4,19} + x_{4,22} + x_{4,24} + x_{4,25} + x_{4,26} = 4$$

$$x_{5,14} + x_{5,15} + x_{5,17} + x_{5,19} + x_{5,22} + x_{5,24} + x_{5,25} + x_{5,26} = 4$$

$$x_{6,14} + x_{6,15} + x_{6,17} + x_{6,19} + x_{6,22} + x_{6,24} + x_{6,25} + x_{6,26} = 4$$

$$x_{7,14} + x_{7,15} + x_{7,17} + x_{7,19} + x_{7,22} + x_{7,24} + x_{7,25} + x_{7,26} = 4$$

$$x_{1,1} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6} + x_{1,11} + x_{1,12} + x_{1,13} = 4$$

$$x_{3,1} + x_{3,3} + x_{3,4} + x_{3,5} + x_{3,6} + x_{3,11} + x_{3,12} + x_{3,13} = 4$$

$$x_{4,1} + x_{4,3} + x_{4,4} + x_{4,5} + x_{4,6} + x_{4,11} + x_{4,12} + x_{4,13} = 4$$

$$x_{5,1} + x_{5,3} + x_{5,4} + x_{5,5} + x_{5,6} + x_{5,11} + x_{5,12} + x_{5,13} = 4$$

$$x_{6,1} + x_{6,3} + x_{6,4} + x_{6,5} + x_{6,6} + x_{6,11} + x_{6,12} + x_{6,13} = 4$$

$$x_{7,1} + x_{7,3} + x_{7,4} + x_{7,5} + x_{7,6} + x_{7,11} + x_{7,12} + x_{7,13} = 4$$

$$x_{1,0} + x_{1,2} + x_{1,7} + x_{1,8} + x_{1,9} + x_{1,10} = 2$$

$$x_{3,0} + x_{3,2} + x_{3,7} + x_{3,8} + x_{3,9} + x_{3,10} = 2$$

$$x_{4,0} + x_{4,2} + x_{4,7} + x_{4,8} + x_{4,9} + x_{4,10} = 2$$

$$x_{5,0} + x_{5,2} + x_{5,7} + x_{5,8} + x_{5,9} + x_{5,10} = 2$$

$$x_{6,0} + x_{6,2} + x_{6,7} + x_{6,8} + x_{6,9} + x_{6,10} = 2$$

$$x_{7,0} + x_{7,2} + x_{7,7} + x_{7,8} + x_{7,9} + x_{7,10} = 2$$

LP solution:

$$D^2 = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . \\
1 & . & 1 & . & . & . & . & 1 & 1 & 1 & 1 & . & . & . & 1 & 1 & . & 1 & . & 1 & . & . & 1 & . & 1 & 1 & 1 & . \\
1 & 1 & . & . & . & . & 1 & . & . & . & 1 & 1 & 1 & . & . & 1 & 1 & . & 1 & . & 1 & . & 1 & 1 & . & 1 & 1 & . \\
. & 1 & . & 1 & . & . & 1 & 1 & . & 1 & . & 1 & . & . & 1 & . & 1 & . & . & 1 & 1 & . & 1 & 1 & . & 1 & . & 1 \\
. & . & 1 & . & 1 & . & . & 1 & . & . & . & 1 & 1 & 1 & . & . & . & 1 & 1 & . & 1 & 1 & 1 & . & 1 & . & 1 & 1 \\
. & . & . & 1 & . & 1 & . & . & 1 & 1 & . & . & 1 & 1 & 1 & . & . & 1 & 1 & . & . & 1 & . & 1 & 1 & 1 & . & 1 \\
. & . & . & . & 1 & 1 & 1 & . & 1 & . & 1 & . & . & 1 & . & 1 & 1 & . & . & 1 & . & 1 & . & 1 & 1 & . & 1 & 1
\end{bmatrix}$$

NRows = 108, Objective = 112, Bicliques remaining = 19.

Iteration 3:

Partition $\Pi^3 = \{\pi_0, \pi_1, \pi_2, \pi_3\}$:

$$\pi_0 = \{3, 4, 5, 13, 21, 27\}$$

$$\pi_1 = \{1, 6, 11, 12, 16, 18, 20, 23\}$$

$$\pi_2 = \{2, 7, 8, 9, 14, 17, 19, 24\}$$

$$\pi_3 = \{0, 10, 15, 22, 25, 26\}$$

Point combination: $L^3 = \{2, 3\}$

Added cuts:

$$x_{0,3} + x_{0,4} + x_{0,5} + x_{0,13} + x_{0,21} + x_{0,27} = 4$$

$$x_{1,3} + x_{1,4} + x_{1,5} + x_{1,13} + x_{1,21} + x_{1,27} = 4$$

$$x_{4,3} + x_{4,4} + x_{4,5} + x_{4,13} + x_{4,21} + x_{4,27} = 4$$

$$x_{5,3} + x_{5,4} + x_{5,5} + x_{5,13} + x_{5,21} + x_{5,27} = 4$$

$$x_{6,3} + x_{6,4} + x_{6,5} + x_{6,13} + x_{6,21} + x_{6,27} = 4$$

$$x_{7,3} + x_{7,4} + x_{7,5} + x_{7,13} + x_{7,21} + x_{7,27} = 4$$

$$x_{0,1} + x_{0,6} + x_{0,11} + x_{0,12} + x_{0,16} + x_{0,18} + x_{0,20} + x_{0,23} = 4$$

$$x_{1,1} + x_{1,6} + x_{1,11} + x_{1,12} + x_{1,16} + x_{1,18} + x_{1,20} + x_{1,23} = 4$$

$$x_{4,1} + x_{4,6} + x_{4,11} + x_{4,12} + x_{4,16} + x_{4,18} + x_{4,20} + x_{4,23} = 4$$

$$x_{5,1} + x_{5,6} + x_{5,11} + x_{5,12} + x_{5,16} + x_{5,18} + x_{5,20} + x_{5,23} = 4$$

$$x_{6,1} + x_{6,6} + x_{6,11} + x_{6,12} + x_{6,16} + x_{6,18} + x_{6,20} + x_{6,23} = 4$$

$$x_{7,1} + x_{7,6} + x_{7,11} + x_{7,12} + x_{7,16} + x_{7,18} + x_{7,20} + x_{7,23} = 4$$

$$x_{0,2} + x_{0,7} + x_{0,8} + x_{0,9} + x_{0,14} + x_{0,17} + x_{0,19} + x_{0,24} = 4$$

$$x_{1,2} + x_{1,7} + x_{1,8} + x_{1,9} + x_{1,14} + x_{1,17} + x_{1,19} + x_{1,24} = 4$$

$$x_{4,2} + x_{4,7} + x_{4,8} + x_{4,9} + x_{4,14} + x_{4,17} + x_{4,19} + x_{4,24} = 4$$

$$x_{5,2} + x_{5,7} + x_{5,8} + x_{5,9} + x_{5,14} + x_{5,17} + x_{5,19} + x_{5,24} = 4$$

$$x_{6,2} + x_{6,7} + x_{6,8} + x_{6,9} + x_{6,14} + x_{6,17} + x_{6,19} + x_{6,24} = 4$$

$$x_{7,2} + x_{7,7} + x_{7,8} + x_{7,9} + x_{7,14} + x_{7,17} + x_{7,19} + x_{7,24} = 4$$

$$x_{0,0} + x_{0,10} + x_{0,15} + x_{0,22} + x_{0,25} + x_{0,26} = 2$$

$$x_{1,0} + x_{1,10} + x_{1,15} + x_{1,22} + x_{1,25} + x_{1,26} = 2$$

$$x_{4,0} + x_{4,10} + x_{4,15} + x_{4,22} + x_{4,25} + x_{4,26} = 2$$

$$x_{5,0} + x_{5,10} + x_{5,15} + x_{5,22} + x_{5,25} + x_{5,26} = 2$$

$$x_{6,0} + x_{6,10} + x_{6,15} + x_{6,22} + x_{6,25} + x_{6,26} = 2$$

$$x_{7,0} + x_{7,10} + x_{7,15} + x_{7,22} + x_{7,25} + x_{7,26} = 2$$

LP solution:

$$D^3 = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . \\
1 & . & 1 & . & . & . & . & 1 & 1 & 1 & 1 & . & . & . & 1 & 1 & . & 1 & . & 1 & . & . & 1 & . & 1 & 1 & 1 & . \\
1 & 1 & . & . & . & . & 1 & . & . & . & 1 & 1 & 1 & . & . & 1 & 1 & . & 1 & . & 1 & . & 1 & 1 & . & 1 & 1 & . \\
. & 1 & . & 1 & . & . & 1 & 1 & . & 1 & . & . & . & 1 & 1 & . & . & . & . & 1 & 1 & 1 & 1 & 1 & . & 1 & . & 1 \\
. & . & 1 & . & 1 & . & . & 1 & . & . & . & 1 & 1 & 1 & . & . & . & 1 & 1 & . & 1 & 1 & 1 & . & 1 & . & 1 & 1 \\
. & . & . & 1 & . & 1 & . & . & 1 & 1 & . & 1 & 1 & . & . & 1 & 1 & 1 & . & . & . & 1 & . & 1 & 1 & 1 & . & 1 \\
. & . & . & . & 1 & 1 & 1 & . & 1 & . & 1 & . & . & 1 & 1 & . & 1 & . & 1 & 1 & . & . & . & 1 & 1 & . & 1 & 1
\end{bmatrix}$$

Rows = 132, Objective = 112, Bicliques remaining = 0.

A design was found at iteration 3. The final LP model has 224 columns and 132 rows. It was solved in a total of 295 simplex iterations, zero B&B nodes, and 0.04 seconds of CPU time on a PC Pentium III, 664 MHz, 256 MB of RAM. In the next chapter, we will include computational results for other instances.

CHAPTER VIII

COMPUTATIONAL IMPLEMENTATION AND RESULTS

In this Chapter we describe the implementation of GDP and include computational results. In Section VIII.1, we propose a separation problem, a *weighted optimal biclique subgraph (WOBS)*, and give and integer programming formulation for it. In Section VIII.2, we compare the analytically derived bounds obtained in Chapter VI with the bounds obtained by solving the WOBS problem and include solution statistics. The computational results confirm our theoretical result that that all the bounds are exact. Also, they show that the WOBS problem solves faster in some separation classes than in others, but in general is consistently fast for all instances. In Section VIII.4, we describe the implementation details of a branch-and-cut algorithm on GDP and include computational results for some block designs.

VIII.1. Separation problem

In principle, any biclique class derived from the original biclique inequalities by complementing or supplementing (like the classes in Tables 13, 16, 18, and 20) can be used in lieu of the original biclique inequalities in GDP to find a t-design. However, as we have show in Chapter VI, some give stronger LP relaxation bounds than others.

The classes of biclique inequalities are combinatorial in size, for example, there are $\binom{v}{t}\binom{b}{\lambda+1}$ biclique subgraphs in the original class of biclique inequalities, and this number grows quickly with the parameters of the design. For this reason, it is impractical to add all the inequalities in the formulation *a priori* and the need for a cutting plane approach or a branch-and-cut algorithm becomes clear. The natural choice of a base formulation (inequalites to start with) for a branch-and-cut approach using GDP are the star inequalities. The biclique inequalities are later added as needed to cut-off infeasible solutions.

To determine which biclique inequalities are unsatisfied by a current LP solution, we could check all the bicliques in the class and return those unsatisfied. As discussed above, this procedure will be computationally intensive and most likely prohibitive as we work with larger parameter designs. As an example, consider the class sizes in Table 22 for a few instances of t-designs and observe how large the class size (i.e. number of induced subgraphs) is as the parameters increase. The column *Nd* gives the number of nonisomorphic designs for the parameters (see [8, 13]). A question mark "?" indicates an open existence problem.

VIII.1.1. Weighted optimal biclique subgraph (WOBS) problem

Given a complete bipartite graph $G = (V \cup B, E) = K_{v,b}$ for GDP formulation with decision variables $x_{ij}, (i, j) = e \in E$. Let $w_{ij} \geq 0$ be a weight assigned to every edge. For the WOBS, we take the relaxation of the edge variables as $x_{ij} \geq 0$ and introduce a new binary variable for each vertex of the bipartite graph G. That is, introduce binary variables:

$$p_i, \qquad i \in V$$

and

$$b_j, \qquad j \in B$$

Separation problem is to find a biclique of size (s,q) with maximum (minimum) weight and to compare with the upper (lower) bound required for that class.

Table 22. Biclique class sizes for some t-designs

t	v	k	λ	b	Nd	Class $K_{t,\lambda+1}$ size	Class $K_{t,\bar{\lambda}+1}$ size
2	6	3	2	10	1	1800	1800
2	7	3	1	7	1	441	735
2	7	3	2	14	4	7644	42042
2	7	3	3	21	10	125685	2441880
2	8	4	3	14	4	28028	28028
2	9	3	1	12	1	2376	33264
2	9	3	2	24	26	72864	89861184
2	9	4	3	18	11	110160	668304
2	10	4	2	15	3	20475	225225
2	11	5	2	11	1	9075	18150
2	13	4	1	13	1	6084	133848
2	13	3	1	26	2	25350	414315330
2	15	3	1	35	80	62475	87617439000
2	15	7	3	15	5	143325	315315
2	16	4	1	20	1	22800	15116400
2	16	6	2	16	3	67200	1372800
2	19	9	4	19	6	67200	1372800
2	25	4	1	50	16	367500	2.81354E+14
2	51	6	1	85	?	4551750	1.53824E+21
2	45	5	1	99	?	4802490	4.24499E+23
2	61	6	1	122	?	13507230	7.49018E+27
2	66	6	1	143	?	21778185	2.39036E+30
2	85	7	1	170	?	51283050	6.17325E+34
2	22	8	4	33	?	54824616	1.89145E+11
2	81	6	1	216	?	75232800	1.59623E+40

Table 22. Continued.

t	v	k	λ	b	Nd	Class $K_{t,\lambda+1}$ size	Class $K_{t,\bar{\lambda}+1}$ size
2	91	7	1	195	?	77456925	1.33325E+38
2	96	6	1	304	?	210015360	2.48062E+51
2	46	10	3	69	?	894758535	7.26912E+21
2	35	7	3	85	?	1204747075	4.97713E+25
2	45	5	2	198	?	1261454040	4.99954E+45
2	70	7	2	230	?	4833525900	2.1629E+50
2	28	7	4	72	?	5288803632	1.07813E+23
2	81	16	3	81	?	5390517600	1.4411E+25
2	55	10	3	99	?	5590098360	2.93204E+29
2	85	15	3	102	?	15170982750	1.08897E+30
2	92	14	3	138	?	60542494740	1.52117E+38
2	70	10	3	161	?	65118736200	6.47719E+42
3	8	4	1	14	1	5096	5096
3	10	4	1	30	1	52200	71253000
3	14	4	1	91	4	1490580	7.19253E+26
3	16	4	1	140	>31300	5448800	3.19923E+42
4	11	5	1	66	1	707850	237837600
4	17	5	1	476	?	269059000	1.9888E+108
5	18	6	1	1428	?	8729746704	3.829E+194
6	19	7	1	3876	?	2.03755E+11	3.3203E+254
7	20	8	1	9690	?	3.63904E+12	2.1339E+245
8	21	9	1	22610	?	5.2011E+13	1.8531E+174
9	22	10	1	49742	?	6.15362E+14	2.02607E+91
10	23	11	1	104006	?	6.18776E+15	2.98773E+37
11	24	12	1	208012	?	5.40026E+16	5.40026E+16

Separation problem can be regarded then as solving a WOBS problem, and can be formulated as the following mixed integer programming problem,

$$
\begin{aligned}
\text{(WOBS)} \qquad \text{Max (Min)} \quad & \sum_i \sum_j w_{ij} x_{ij} && && (8.1)\\
\text{subject to} \quad & x_{ij} - p_i \le 0, && i \in V, j \in B && (8.2)\\
& x_{ij} - b_j \le 0, && j \in B, i \in V && (8.3)\\
& p_i + b_j - x_{i,j} \le 1, && i \in V, j \in B && (8.4)\\
& \sum_i p_i \le (\ge)\ s, && && (8.5)\\
& \sum_j b_j \le (\ge)\ q, && && (8.6)\\
& x_{ij} \ge 0, \quad p_i \in \{0,1\}, b_j \in \{0,1\}.
\end{aligned}
$$

The weights w_{ij} correspond to the current LP relaxation solution. If the maximum (minimum) value of the WOBS problem falls outside the computed bound for a particular class, then the corresponding biclique inequality is added to the formulation at that particular node of the branch-and-bound tree. If there are still violated cuts, the separation problem is solved again and cuts are added until no more unsatisfied inequalities are found, then the LP is re-optimized.

There are some observations that would make this process more efficient. When solving the separation problem we are looking for a biclique subgraph $K_{s,q}$ such that the sum of the weights of its edges is maximum (minimum) and then compare this sum to the required computed upper (lower) bound. The maximization separation problem does not need to be solved if, at any iteration, the sum of the sq largest edge weights (regardless if they correspond to a biclique graph) is at or below the upper bound. Similarly, there is no need to solve the minimization separation problem if the sum of the smallest sq edge weights is at or above the lower bound. This observations prevent waste of computational time in setting up and solving the WOBS problem unnecessarily.

VIII.2. Comparison of derived bounds versus actual bounds

We test computationally the exactness of the bounds of the derived cuts by comparing the the actual bounds that known t-designs yield. The standard test instances were obtained form Mathon and Rosa [40]. Those are block designs, each with its corresponding number of nonisomorphic instances, except for 2-(15,3,1) for which we used only the four resolvable instances out of the 80 possible, and for the 2-(25,4,1) for which we only used one instance out of the 16 possible.

The bounds results were obtained by solving a WOBS problem independently for the corresponding biclique size. That is, we solved 95 instances for 8 classes and for lower and upper bound for each class. Therefore, the total mixed integer programming problems (8.1) solved were 1520. The results for the bounds are divided in two tables, Table 23 corresponding to the classes using parameter λ, and Table 24 corresponding to the classes using parameter $\overline{\lambda}$. All the bounds resulted exact (% error equal zero) with respect to the analytical derivations from Tables 13, 16, 18 and 20. The analytically computed bounds are presented in Tables 25 and 26. The solution statistics for the WOBS solved are presented in Tables 27, 28, 29 and 30. The biclique sizes are selected accordingly to the classes derived in Chapter VI. The headers are as follows: *Nd* refers to the number of nonisomorphic designs [40]; *MIP size* is the number of columns and rows in the WOBS problem; *ite* is the number of Simplex iterations; *bbn* is the number of branch-and-bound nodes solved; and *sec* is the CPU time in seconds on a PC Pentium III, 664 MHz, 254 MB of RAM.

Table 23. Actual bounds by solving WOBS problem for classes with parameter λ

Design						$\overline{\lambda}$	Nd	Classes Bounds			
t	v	k	λ	b	r			$K_{t,\lambda+1}$	$K_{v-t,\lambda+1}$	$K_{t,b-(\lambda+1)}$	$K_{v-t,b-(\lambda+1)}$
2	6	3	2	10	5	2	1	[1,5]	[4,8]	[5,9]	[12,16]
2	7	3	1	7	3	2	1	[0,3]	[3,6]	[3,6]	[9,12]
2	7	3	2	14	6	4	1	[0,5]	[4,9]	[7,12]	[21,26]
							2	[0,5]	[4,9]	[7,12]	[21,26]
							3	[0,5]	[4,9]	[7,12]	[21,26]
							4	[0,5]	[4,9]	[7,12]	[21,26]
2	7	3	3	21	9	6	1	[0,7]	[5,12]	[11,18]	[33,40]
							2	[0,7]	[5,12]	[11,18]	[33,40]
							3	[0,7]	[5,12]	[11,18]	[33,40]
							4	[0,7]	[5,12]	[11,18]	[33,40]
							5	[0,7]	[5,12]	[11,18]	[33,40]
							6	[0,7]	[5,12]	[11,18]	[33,40]
							7	[0,7]	[5,12]	[11,18]	[33,40]
							8	[0,7]	[5,12]	[11,18]	[33,40]
							9	[0,7]	[5,12]	[11,18]	[33,40]
							10	[0,7]	[5,12]	[11,18]	[33,40]
2	8	4	3	14	7	3	1	[1,7]	[9,15]	[7,13]	[27,33]
							2	[1,7]	[9,15]	[7,13]	[27,33]
							3	[1,7]	[9,15]	[7,13]	[27,33]
							4	[1,7]	[9,15]	[7,13]	[27,33]
2	9	3	1	12	4	5	1	[0,3]	[3,6]	[5,8]	[22,25]
2	9	3	2	24	8	10	1	[0,5]	[4,9]	[11,16]	[47,52]
							2	[0,5]	[4,9]	[11,16]	[47,52]
							3	[0,5]	[4,9]	[11,16]	[47,52]
							4	[0,5]	[4,9]	[11,16]	[47,52]
							5	[0,5]	[4,9]	[11,16]	[47,52]
							6	[0,5]	[4,9]	[11,16]	[47,52]
							7	[0,5]	[4,9]	[11,16]	[47,52]
							8	[0,5]	[4,9]	[11,16]	[47,52]
							9	[0,5]	[4,9]	[11,16]	[47,52]
							10	[0,5]	[4,9]	[11,16]	[47,52]
							11	[0,5]	[4,9]	[11,16]	[47,52]
							12	[0,5]	[4,9]	[11,16]	[47,52]
							13	[0,5]	[4,9]	[11,16]	[47,52]

Table 23. Continued

Design								Classes Bounds			
t	v	k	λ	b	r	$\overline{\lambda}$	Nd	$K_{t,\lambda+1}$	$K_{v-t,\lambda+1}$	$K_{t,b-(\lambda+1)}$	$K_{v-t,b-(\lambda+1)}$
2	9	3	2	24	8	10	14	[0,5]	[4,9]	[11,16]	[47,52]
							15	[0,5]	[4,9]	[11,16]	[47,52]
							16	[0,5]	[4,9]	[11,16]	[47,52]
							17	[0,5]	[4,9]	[11,16]	[47,52]
							18	[0,5]	[4,9]	[11,16]	[47,52]
							19	[0,5]	[4,9]	[11,16]	[47,52]
							20	[0,5]	[4,9]	[11,16]	[47,52]
							21	[0,5]	[4,9]	[11,16]	[47,52]
							22	[0,5]	[4,9]	[11,16]	[47,52]
							23	[0,5]	[4,9]	[11,16]	[47,52]
							24	[0,5]	[4,9]	[11,16]	[47,52]
							25	[0,5]	[4,9]	[11,16]	[47,52]
							26	[0,5]	[4,9]	[11,16]	[47,52]
							27	[0,5]	[4,9]	[11,16]	[47,52]
							28	[0,5]	[4,9]	[11,16]	[47,52]
							29	[0,5]	[4,9]	[11,16]	[47,52]
							30	[0,5]	[4,9]	[11,16]	[47,52]
							31	[0,5]	[4,9]	[11,16]	[47,52]
							32	[0,5]	[4,9]	[11,16]	[47,52]
							33	[0,5]	[4,9]	[11,16]	[47,52]
							34	[0,5]	[4,9]	[11,16]	[47,52]
							35	[0,5]	[4,9]	[11,16]	[47,52]
							36	[0,5]	[4,9]	[11,16]	[47,52]
2	9	4	3	18	8	5	1	[0,7]	[9,16]	[9,16]	[40,7]
							2	[0,7]	[9,16]	[9,16]	[40,7]
							3	[0,7]	[9,16]	[9,16]	[40,7]
							4	[0,7]	[9,16]	[9,16]	[40,7]
							5	[0,7]	[9,16]	[9,16]	[40,7]
							6	[0,7]	[9,16]	[9,16]	[40,7]
							7	[0,7]	[9,16]	[9,16]	[40,7]
							8	[0,7]	[9,16]	[9,16]	[40,7]
							9	[0,7]	[9,16]	[9,16]	[40,7]
							10	[0,7]	[9,16]	[9,16]	[40,7]
							11	[0,7]	[9,16]	[9,16]	[40,7]

Table 23. Continued

Design								Classes Bounds			
t	v	k	λ	b	r	$\overline{\lambda}$	Nd	$K_{t,\lambda+1}$	$K_{v-t,\lambda+1}$	$K_{t,b-(\lambda+1)}$	$K_{v-t,b-(\lambda+1)}$
2	10	4	2	15	6	5	1	[0,5]	[7,12]	[7,12]	[36,41]
							2	[0,5]	[7,12]	[7,12]	[36,41]
							3	[0,5]	[7,12]	[7,12]	[36,41]
2	11	5	2	11	5	3	1	[0,5]	[10,15]	[5,10]	[30,35]
2	13	3	1	26	6	15	1	[0,3]	[3,6]	[5,10]	[30,35]
							2	[0,3]	[3,6]	[5,10]	[30,35]
2	13	4	1	13	4	6	1	[0,3]	[5,8]	[5,8]	[36,39]
2	15	3	1	35	7	22	1	[0,3]	[3,6]	[11,14]	[85,88]
							2	[0,3]	[3,6]	[11,14]	[85,88]
							3	[0,3]	[3,6]	[11,14]	[85,88]
							4	[0,3]	[3,6]	[11,14]	[85,88]
2	15	7	3	15	7	4	1	[0,7]	[21,28]	[7,14]	[63,70]
							2	[0,7]	[21,28]	[7,14]	[63,70]
							3	[0,7]	[21,28]	[7,14]	[63,70]
							4	[0,7]	[21,28]	[7,14]	[63,70]
							5	[0,7]	[21,28]	[7,14]	[63,70]
2	16	4	1	20	5	11	1	[0,3]	[5,8]	[7,10]	[62,65]
2	16	6	2	16	6	6	1	[0,5]	[13,18]	[7,12]	[66,71]
							2	[0,5]	[13,18]	[7,12]	[66,71]
							3	[0,5]	[13,18]	[7,12]	[66,71]
2	19	9	4	19	9	5	1	[0,9]	[36,45]	[9,18]	[108,117]
							2	[0,9]	[36,45]	[9,18]	[108,117]
							3	[0,9]	[36,45]	[9,18]	[108,117]
							4	[0,9]	[36,45]	[9,18]	[108,117]
							5	[0,9]	[36,45]	[9,18]	[108,117]
							6	[0,9]	[36,45]	[9,18]	[108,117]
2	25	4	1	50	8	35	1	[0,3]	[5,8]	[13,16]	[176,179]

Table 24. Actual bounds by solving WOBS problem for other classes with parameter $\overline{\lambda}$

Design								Classes Bounds			
t	v	k	λ	b	r	$\overline{\lambda}$	Nd	$K_{t,\overline{\lambda}+1}$	$K_{v-t,\overline{\lambda}+1}$	$K_{t,b-(\overline{\lambda}+1)}$	$K_{v-t,b-(\overline{\lambda}+1)}$
2	6	3	2	10	5	2	1	[1,5]	[4,8]	[5,9]	[12,16]
2	7	3	1	7	3	2	1	[1,4]	[5,8]	[2,5]	[7,10]
2	7	3	2	14	6	4	1	[1,7]	[8,14]	[5,11]	[16,22]
							2	[1,7]	[8,14]	[5,11]	[16,22]
							3	[1,7]	[8,14]	[5,11]	[16,22]
							4	[1,7]	[8,14]	[5,11]	[16,22]
2	7	3	3	21	9	6	1	[1,10]	[11,20]	[8,17]	[25,34]
							2	[1,10]	[11,20]	[8,17]	[25,34]
							3	[1,10]	[11,20]	[8,17]	[25,34]
							4	[1,10]	[11,20]	[8,17]	[25,34]
							5	[1,10]	[11,20]	[8,17]	[25,34]
							6	[1,10]	[11,20]	[8,17]	[25,34]
							7	[1,10]	[11,20]	[8,17]	[25,34]
							8	[1,10]	[11,20]	[8,17]	[25,34]
							9	[1,10]	[11,20]	[8,17]	[25,34]
							10	[1,10]	[11,20]	[8,17]	[25,34]
2	8	4	3	14	7	3	1	[1,7]	[9,15]	[7,13]	[27,33]
							2	[1,7]	[9,15]	[7,13]	[27,33]
							3	[1,7]	[9,15]	[7,13]	[27,33]
							4	[1,7]	[9,15]	[7,13]	[27,33]
2	9	3	1	12	4	5	1	[1,7]	[11,17]	[1,7]	[11,17]
2	9	3	2	24	8	10	1	[1,13]	[20,32]	[3,15]	[24,36]
							2	[1,13]	[20,32]	[3,15]	[24,36]
							3	[1,13]	[20,32]	[3,15]	[24,36]
							4	[1,13]	[20,32]	[3,15]	[24,36]
							5	[1,13]	[20,32]	[3,15]	[24,36]
							6	[1,13]	[20,32]	[3,15]	[24,36]
							7	[1,13]	[20,32]	[3,15]	[24,36]
							8	[1,13]	[20,32]	[3,15]	[24,36]
							9	[1,13]	[20,32]	[3,15]	[24,36]
							10	[1,13]	[20,32]	[3,15]	[24,36]
							11	[1,13]	[20,32]	[3,15]	[24,36]
							12	[1,13]	[20,32]	[3,15]	[24,36]
							13	[1,13]	[20,32]	[3,15]	[24,36]

Table 24. Continued

Design								Classes Bounds			
t	v	k	λ	b	r	$\overline{\lambda}$	Nd	$K_{t,\lambda+1}$	$K_{v-t,\lambda+1}$	$K_{t,b-(\lambda+1)}$	$K_{v-t,b-(\lambda+1)}$
2	9	3	2	24	8	10	14	[1,13]	[20,32]	[3,15]	[24,36]
							15	[1,13]	[20,32]	[3,15]	[24,36]
							16	[1,13]	[20,32]	[3,15]	[24,36]
							17	[1,13]	[20,32]	[3,15]	[24,36]
							18	[1,13]	[20,32]	[3,15]	[24,36]
							19	[1,13]	[20,32]	[3,15]	[24,36]
							20	[1,13]	[20,32]	[3,15]	[24,36]
							21	[1,13]	[20,32]	[3,15]	[24,36]
							22	[1,13]	[20,32]	[3,15]	[24,36]
							23	[1,13]	[20,32]	[3,15]	[24,36]
							24	[1,13]	[20,32]	[3,15]	[24,36]
							25	[1,13]	[20,32]	[3,15]	[24,36]
							26	[1,13]	[20,32]	[3,15]	[24,36]
							27	[1,13]	[20,32]	[3,15]	[24,36]
							28	[1,13]	[20,32]	[3,15]	[24,36]
							29	[1,13]	[20,32]	[3,15]	[24,36]
							30	[1,13]	[20,32]	[3,15]	[24,36]
							31	[1,13]	[20,32]	[3,15]	[24,36]
							32	[1,13]	[20,32]	[3,15]	[24,36]
							33	[1,13]	[20,32]	[3,15]	[24,36]
							34	[1,13]	[20,32]	[3,15]	[24,36]
							35	[1,13]	[20,32]	[3,15]	[24,36]
							36	[1,13]	[20,32]	[3,15]	[24,36]
9	4	3	18	8	5	1	1	[1,9]	[15,23]	[7,15]	[33,41]
							2	[1,9]	[15,23]	[7,15]	[33,41]
							3	[1,9]	[15,23]	[7,15]	[33,41]
							4	[1,9]	[15,23]	[7,15]	[33,41]
							5	[1,9]	[15,23]	[7,15]	[33,41]
							6	[1,9]	[15,23]	[7,15]	[33,41]
							7	[1,9]	[15,23]	[7,15]	[33,41]
							8	[1,9]	[15,23]	[7,15]	[33,41]
							9	[1,9]	[15,23]	[7,15]	[33,41]
							10	[1,9]	[15,23]	[7,15]	[33,41]
							11	[1,9]	[15,23]	[7,15]	[33,41]

Table 24. Continued

Design								Classes Bounds			
t	v	k	λ	b	r	$\overline{\lambda}$	Nd	$K_{t,\lambda+1}$	$K_{v-t,\lambda+1}$	$K_{t,b-(\lambda+1)}$	$K_{v-t,b-(\lambda+1)}$
2	10	4	2	15	6	5	1	[1,8]	[16,23]	[4,11]	[25,32]
							2	[1,8]	[16,23]	[4,11]	[25,32]
							2	[1,8]	[16,23]	[4,11]	[25,32]
2	11	5	2	11	5	3	1	[1,6]	[14,19]	[4,9]	[26,31]
2	13	3	1	26	6	15	1	[1,12]	[36,47]	[0,11]	[19,30]
							2	[1,12]	[36,47]	[0,11]	[19,30]
2	13	4	1	13	4	6	1	[1,8]	[20,27]	[0,7]	[17,24]
2	15	3	1	35	7	22	1	[1,14]	[55,68]	[0,13]	[23,36]
							2	[1,14]	[55,68]	[0,13]	[23,36]
							3	[1,14]	[55,68]	[0,13]	[23,36]
							4	[1,14]	[55,68]	[0,13]	[23,36]
2	15	7	3	15	7	4	1	[1,8]	[27,34]	[6,13]	[57,64]
							2	[1,8]	[27,34]	[6,13]	[57,64]
							3	[1,8]	[27,34]	[6,13]	[57,64]
							4	[1,8]	[27,34]	[6,13]	[57,64]
							5	[1,8]	[27,34]	[6,13]	[57,64]
2	16	4	1	20	5	11	1	[1,10]	[38,47]	[0,9]	[23,32]
2	16	6	2	16	6	6	1	[1,9]	[33,41]	[3,11]	[43,51]
							2	[1,9]	[33,41]	[3,11]	[43,51]
							3	[1,9]	[33,41]	[3,11]	[43,51]
2	19	9	4	19	9	5	1	[1,10]	[44,53]	[8,17]	[100,109]
							2	[1,10]	[44,53]	[8,17]	[100,109]
							3	[1,10]	[44,53]	[8,17]	[100,109]
							4	[1,10]	[44,53]	[8,17]	[100,109]
							5	[1,10]	[44,53]	[8,17]	[100,109]
							6	[1,10]	[44,53]	[8,17]	[100,109]
2	25	4	1	50	8	35	1	[1,16]	[128,143]	[0,15]	[41,56]

Table 25. Analytically computed bounds for classes with parameter λ

Design							$K_{t,\lambda+1}$		$K_{v-t,\lambda+1}$		$K_{t,b-(\lambda+1)}$		$K_{v-t,b-(\lambda+1)}$		
t	v	k	λ	b	r	$\overline{\lambda}$	lb	ub	lb	ub	lb	ub	lb	ub	% Error
2	6	3	2	10	5	2	1	5	4	8	5	9	12	16	0
2	7	3	1	7	3	2	0	3	3	6	3	6	9	12	0
2	7	3	2	14	6	4	0	5	4	9	7	12	21	26	0
2	7	3	3	21	9	6	0	7	5	12	11	18	33	40	0
2	8	4	3	14	7	3	1	7	9	15	7	13	27	33	0
2	9	3	1	12	4	5	0	3	3	6	5	8	22	25	0
2	9	3	2	24	8	10	0	5	4	9	11	16	47	52	0
2	9	4	3	18	8	5	0	7	9	16	9	16	40	47	0
2	10	4	2	15	6	5	0	5	7	12	7	12	36	41	0
2	11	5	2	11	5	3	0	5	10	15	5	10	30	35	0
2	13	3	1	26	6	15	0	3	3	6	9	12	60	63	0
2	13	4	1	13	4	6	0	3	5	8	5	8	36	39	0
2	15	3	1	35	7	22	0	3	3	6	11	14	85	88	0
2	15	7	3	15	7	4	0	7	21	28	7	14	63	70	0
2	16	4	1	20	5	11	0	3	5	8	7	10	62	65	0
2	16	6	2	16	6	6	0	5	13	18	7	12	66	71	0
2	19	9	4	19	9	5	0	9	36	45	9	18	108	117	0
2	25	4	1	50	8	35	0	3	5	8	13	16	176	179	0

Table 26. Analytically computed bounds for other classes with parameter $\overline{\lambda}$

Design							$K_{t,\overline{\lambda}+1}$		$K_{v-t,\overline{\lambda}+1}$		$K_{t,b-(\overline{\lambda}+1)}$		$K_{v-t,b-(\overline{\lambda}+1)}$		
t	v	k	λ	b	r	$\overline{\lambda}$	lb	ub	lb	ub	lb	ub	lb	ub	% Error
2	6	3	2	10	5	2	1	5	4	8	5	9	12	16	0
2	7	3	1	7	3	2	1	4	5	8	2	5	7	10	0
2	7	3	2	14	6	4	1	7	8	14	5	11	16	22	0
2	7	3	3	21	9	6	1	10	11	20	8	17	25	34	0
2	8	4	3	14	7	3	1	7	9	15	7	13	27	33	0
2	9	3	1	12	4	5	1	7	11	17	1	7	11	17	0
2	9	3	2	24	8	10	1	13	20	32	3	15	24	36	0
2	9	4	3	18	8	5	1	9	15	23	7	15	33	41	0
2	10	4	2	15	6	5	1	8	16	23	4	11	25	32	0
2	11	5	2	11	5	3	1	6	14	19	4	9	26	31	0
2	13	3	1	26	6	15	1	12	36	47	0	11	19	30	0
2	13	4	1	13	4	6	1	8	20	27	0	7	17	24	0
2	15	3	1	35	7	22	1	14	55	68	0	13	23	36	0
2	15	7	3	15	7	4	1	8	27	34	6	13	57	64	0
2	16	4	1	20	5	11	1	10	38	47	0	9	23	32	0
2	16	6	2	16	6	6	1	9	33	41	3	11	43	51	0
2	19	9	4	19	9	5	1	10	44	53	8	17	100	109	0
2	25	4	1	50	8	35	1	16	128	143	0	15	41	56	0

Table 27. Statistics on maximize WOBS for classes with parameter λ

Design	MIP size			$K_{t,\lambda+1}$			$K_{v-t,\lambda+1}$			$K_{t,b-(\lambda+1)}$			$K_{v-t,b-(\lambda+1)}$		
t-(v, k, λ)	Col	Row	Nd	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec
2-(6,3,2)	76	182	1	882	167	0.26	579	105	0.16	245	29	0.04	538	91	0.15
2-(7,3,1)	63	149	1	342	85	0.11	66	5	0.01	70	6	0.01	306	59	0.1
2-(7,3,2)	119	296	1	2373	397	0.93	158	7	0.03	141	6	0.03	905	159	0.37
			2	2003	347	0.81	125	11	0.04	136	6	0.03	879	129	0.33
			3	2147	349	0.86	118	7	0.02	134	6	0.02	981	185	0.43
			4	2389	394	0.94	133	8	0.03	141	6	0.02	760	97	0.27
2-(7,3,3)	175	443	1	6673	901	3.33	152	9	0.04	200	6	0.03	897	47	0.25
			2	5127	649	2.35	212	13	0.06	201	6	0.04	1130	111	0.46
			3	4637	635	2.27	175	11	0.05	198	6	0.03	1054	95	0.41
			4	4325	525	1.95	344	24	0.11	202	6	0.04	976	61	0.3
			5	5454	719	3.43	169	10	0.05	204	6	0.04	902	53	0.28
			6	5870	737	2.68	179	18	0.07	207	6	0.04	1002	61	0.31
			7	5088	653	2.36	207	12	0.05	204	6	0.03	1005	73	0.34
			8	5127	631	2.32	307	24	0.1	203	6	0.04	970	89	0.38
			9	4996	673	2.4	319	26	0.11	198	6	0.03	971	65	0.32
			10	4925	669	2.38	158	9	0.05	203	6	0.04	934	77	0.36
2-(8,4,3)	134	338	1	5884	913	2.56	2771	452	1.2	558	55	0.17	3982	596	1.79
			2	6380	974	2.83	2240	381	1.01	567	55	0.17	4176	644	1.9
			3	6417	1019	2.92	2727	458	1.26	592	55	0.17	2972	437	1.31
			4	6461	1009	2.93	2264	449	1.17	539	55	0.17	3383	509	1.48
2-(9,3,1)	129	326	1	1206	245	0.6	113	7	0.02	130	8	0.03	996	171	0.46
2-(9,3,2)	249	650	1	7515	894	4.47	150	8	0.05	216	8	0.06	1697	233	1.16
			2	7282	1025	4.94	149	7	0.05	225	8	0.06	1478	133	0.76
			3	7355	1022	5.09	166	13	0.08	220	8	0.06	1458	103	0.66
			4	7852	1089	5.28	173	12	0.07	211	8	0.06	1393	83	0.54
			5	7552	1070	5.32	203	18	0.1	233	8	0.06	1540	133	0.78
			6	7371	1025	5.05	196	8	0.05	210	8	0.06	1328	77	0.52
			7	7733	1069	5.4	177	13	0.07	227	8	0.06	1532	131	0.76
			8	8389	1241	6.08	157	12	0.07	227	8	0.06	1447	133	0.77
			9	7014	977	4.82	158	13	0.08	220	8	0.06	1414	111	0.65
			10	7775	1039	5.13	177	16	0.08	227	8	0.06	1654	163	0.91
			11	7496	971	4.85	168	15	0.08	231	8	0.06	1698	227	1.12
			12	7135	886	4.4	154	12	0.07	219	8	0.05	1554	139	0.8
			13	7201	963	4.71	167	13	0.07	226	8	0.06	1485	103	0.65
			14	7530	1008	4.94	172	13	0.07	225	8	0.06	1518	97	0.64
			15	7066	967	4.73	170	14	0.07	233	8	0.06	1470	115	0.69
			16	7627	1067	5.15	163	13	0.07	226	8	0.06	1492	137	0.77
			17	6823	953	4.65	152	7	0.04	230	8	0.05	1534	131	0.76
			18	8065	1063	5.42	182	14	0.07	229	8	0.06	1539	101	0.66
			19	6856	803	4.3	188	14	0.08	239	8	0.06	1758	253	1.25
			20	7862	1101	5.63	183	16	0.09	212	7	0.06	1740	249	1.27
			21	7240	1110	5.4	186	12	0.08	217	8	0.06	1514	133	0.77
			22	7523	1032	5.08	152	11	0.06	211	8	0.06	1507	135	0.77
			23	7999	1141	5.6	160	13	0.07	230	8	0.06	1475	141	0.8
			24	7910	1097	5.49	165	11	0.06	210	8	0.05	1500	141	0.81
			25	8214	1143	5.67	157	12	0.07	214	8	0.06	1568	133	0.8
			26	6958	965	4.76	217	20	0.11	223	8	0.06	1495	111	0.69
			27	6316	771	4.03	171	14	0.08	225	8	0.06	1450	133	0.77

Table 27. Continued

Design	MIP size			$K_{t,\lambda+1}$			$K_{v-t,\lambda+1}$			$K_{t,b-(\lambda+1)}$			$K_{v-t,b-(\lambda+1)}$		
t-(v, k, λ)	Col	Row	Nd	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec
2-(9,3,2)	28	6563	913	4.51	141	12	0.07	221	8	0.06	1390	103	0.66		
			29	7179	939	4.71	165	12	0.07	221	8	0.06	1820	269	1.37
			30	6575	890	4.67	162	13	0.07	222	8	0.06	1503	103	0.65
			31	7121	957	4.81	155	11	0.06	219	8	0.05	1471	111	0.69
			32	7317	976	4.95	168	12	0.07	219	8	0.06	1548	103	0.7
			33	7238	989	4.99	162	12	0.06	228	8	0.06	1403	103	0.62
			34	6863	935	4.59	156	12	0.06	220	8	0.05	1420	105	0.65
			35	7287	1055	5.07	163	12	0.06	206	8	0.05	1404	131	0.75
			36	7017	948	4.62	169	12	0.06	234	8	0.06	1527	103	0.65
2-(9,4,3)	189	488	1	15532	1985	7.97	183	11	0.06	197	8	0.04	9185	1133	4.85
			2	13782	1829	7.13	184	10	0.06	222	8	0.05	7332	911	3.97
			3	15505	2051	8.26	241	14	0.07	220	8	0.05	8020	1007	4.28
			4	16216	2113	8.47	182	11	0.05	220	8	0.05	7877	999	4.15
			5	12582	1657	6.61	593	36	0.18	204	8	0.05	7362	931	3.95
			6	14558	1891	7.54	183	11	0.06	225	8	0.05	7167	921	3.89
			7	15007	1973	7.83	444	27	0.13	208	8	0.05	6734	811	3.47
			8	14850	1863	7.54	184	10	0.05	207	8	0.05	7170	921	3.9
			9	14320	1895	7.5	188	10	0.05	200	8	0.04	7509	945	4.02
			10	13149	1667	6.7	491	32	0.16	200	8	0.05	8712	1197	4.96
			11	15070	1945	7.77	187	11	0.06	202	8	0.06	7993	983	4.18
2-(10,4,2)	175	452	1	8032	1269	4.59	197	7	0.03	182	9	0.05	4532	653	2.56
			2	7655	1104	4.09	143	4	0.03	174	9	0.05	4907	755	2.94
			3	8063	1215	4.49	156	6	0.04	175	8	0.04	5629	799	3.17
2-(11,5,2)	143	365	1	4417	825	2.27	172	7	0.03	151	10	0.04	3472	579	1.79
2-(13,3,1)	377	1016	1	7687	1101	8.6	210	17	0.14	225	12	0.12	2570	167	1.79
			2	7690	1121	8.7	190	17	0.13	231	12	0.11	2530	187	1.92
2-(13,4,1)	195	509	1	3725	655	2.67	134	7	0.04	145	12	0.06	3387	501	2.3
2-(15,3,1)	575	1577	1	13191	1297	16.91	279	24	0.33	300	14	0.22	4449	209	4.38
			2	15166	1849	23.48	250	25	0.33	292	13	0.21	4104	237	4.4
			3	14433	1597	20.79	273	27	0.37	288	13	0.22	4290	203	4.12
			4	15391	2011	25.33	248	25	0.34	279	11	0.19	4325	237	4.49
2-(15,7,3)	255	677	1	46623	6127	31.93	340	13	0.1	261	13	0.08	35062	4077	22.94
			2	44929	5985	31.33	310	11	0.08	253	13	0.09	34671	4137	23.24
			3	45964	5941	31.14	344	13	0.11	252	13	0.08	34343	4157	23.72
			4	45116	5975	31.52	339	13	0.1	248	13	0.08	33158	3959	22.06
			5	44804	5961	30.38	338	13	0.1	253	12	0.08	35356	4097	23.35
2-(16,4,1)	356	962	1	12985	1503	11.38	210	16	0.13	203	12	0.11	11745	1053	9.7
2-(16,6,2)	288	770	1	29255	3591	22.36	307	12	0.1	238	15	0.11	26245	2877	19.42
			2	29006	3478	21.18	294	13	0.12	237	15	0.1	25780	2921	19.35
			3	29403	3775	22.74	303	14	0.13	236	15	0.11	27192	2973	20.09
2-(19,9,4)	399	1085	1	414265	50275	440.7	888	41	0.48	338	10	0.13	269959	24603	244.57
			2	416350	50861	444.09	1035	34	0.46	338	10	0.12	271770	25397	253.38
			3	417181	50773	438.87	855	25	0.37	368	12	0.15	267622	24324	240.65
			4	414059	50359	437.12	1418	57	0.71	359	14	0.16	275002	25328	249.43
			5	417855	50615	444.1	1026	43	0.54	404	13	0.16	263547	24365	245.35
			6	415785	50319	437.54	1238	47	0.61	329	10	0.13	278117	27101	262.86
2-(25,4,1)	1325	3752	1	179420	13063	555.13	497	34	1.34	412	15	0.7	28483	707	78.91

Table 28. Statistics on maximize WOBS for other classes with parameter $\overline{\lambda}$

Design	MIP size			$K_{t,\overline{\lambda}+1}$			$K_{v-t,\overline{\lambda}+1}$			$K_{t,b-(\overline{\lambda}+1)}$			$K_{v-t,b-(\overline{\lambda}+1)}$		
t-(v, k, λ)	Col	Row	Nd	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec
2-(6,3,2)	76	182	1	882	167	0.30	579	105	0.16	245	29	0.04	538	91	0.14
2-(7,3,1)	63	149	1	388	101	0.13	369	81	0.11	270	51	0.07	404	100	0.14
2-(7,3,2)	119	296	1	3841	719	1.7	2225	347	0.84	365	35	0.1	2326	349	0.88
			2	4109	787	1.84	2383	331	0.84	398	41	0.11	2280	397	0.98
			3	4145	777	1.84	1864	307	0.74	385	39	0.11	3082	659	1.52
			4	3296	655	1.55	1821	303	0.72	384	41	0.1	2238	387	0.94
2-(7,3,3)	175	443	1	34073	4813	17.24	5434	790	2.8	523	37	0.16	7174	909	3.55
			2	38186	5719	20.27	5735	846	3.05	542	41	0.18	8131	1421	5.08
			3	39648	6131	21.72	7801	1046	3.86	534	41	0.16	5247	883	3.19
			4	35810	5473	19.55	6212	973	3.46	516	41	0.26	3637	533	2.06
			5	39553	6053	21.21	6804	963	3.53	525	41	0.16	6774	1349	4.58
			6	34228	4909	17.19	6703	955	3.38	542	39	0.17	9316	1793	5.88
			7	39500	5869	20.83	7093	1078	3.79	517	41	0.17	6976	1157	4.07
			8	36975	5801	20.11	8234	1078	3.9	542	41	0.18	5930	1059	3.75
			9	32304	4649	16.6	6062	843	3.28	533	41	0.17	5578	919	3.34
			10	34555	5205	18.99	7321	1011	3.65	649	55	0.23	5777	1037	3.71
2-(8,4,3)	134	338	1	5884	913	2.62	2771	452	1.22	558	55	0.16	3982	596	1.76
			2	6380	974	2.81	2240	381	1.02	567	55	0.17	4176	644	1.9
			3	6417	1019	2.98	2727	458	1.23	592	55	0.17	2972	437	1.28
			4	6461	1009	2.93	2264	449	1.17	539	55	0.17	3383	509	1.5
2-(9,3,1)	129	326	1	787	119	0.3	1621	273	0.72	787	119	0.3	1621	273	0.72
2-(9,3,2)	249	650	1	2042	301	1.41	20833	2261	11.96	895	69	0.36	48608	5522	29.01
			2	3746	697	3.16	18152	2108	11.36	871	71	0.4	52713	6107	32.91
			3	2494	463	2.11	16008	1680	9.19	845	69	0.37	48954	5185	28.45
			4	2786	511	2.34	20588	2084	11.33	854	71	0.39	50196	5860	31.09
			5	1684	247	1.15	17270	1980	10.65	852	69	0.37	37163	4584	23.76
			6	3713	715	3.2	17875	1891	10.52	839	71	0.38	50526	5845	31.32
			7	2786	479	2.2	19087	2036	11.31	944	71	0.42	41475	4881	25.89
			8	4207	679	3.18	14581	1618	8.83	863	71	0.38	55487	6301	34.01
			9	3658	665	2.99	20127	2285	12.41	896	71	0.41	59571	7113	37.13
			10	3446	633	2.92	20258	2102	11.52	821	71	0.38	59347	7060	37.97
			11	2283	351	1.66	17348	1824	10.04	944	71	0.4	47228	5278	27.9
			12	3375	623	2.77	18961	2169	11.79	834	71	0.38	64769	7489	39.73
			13	2333	379	1.75	21089	2124	11.8	854	71	0.39	47607	5353	28.57
			14	2695	469	2.14	19950	2023	11.36	859	71	0.39	61221	7042	37.13
			15	2745	443	2.08	22285	2327	12.73	921	71	0.4	54951	6544	34.7
			16	1673	221	1.06	24369	2710	14.52	873	71	0.39	51829	5684	30.94
			17	4216	765	3.55	14300	1719	9.25	872	71	0.39	46814	5153	27.8
			18	2860	469	2.24	22004	2382	13.05	896	71	0.4	61178	7131	38.55
			19	5005	759	3.73	24847	2638	14.6	967	77	0.43	60038	6687	37.01
			20	2355	363	1.76	16791	1806	9.81	909	69	0.39	49801	5833	31.13
			21	2508	397	1.85	19512	2062	11.52	879	71	0.39	44850	4991	26.65
			22	3888	645	3.02	20000	2175	11.96	862	71	0.4	61561	7182	39.03
			23	3089	593	2.76	16625	1680	9.63	894	71	0.41	64624	7403	39.32
			24	3348	643	3.01	19433	2078	11.63	774	71	0.38	60207	6578	36.15
			25	2889	473	2.28	21420	2338	12.87	884	71	0.41	59852	7031	38.57
			26	2591	429	2.01	20226	2354	13.18	813	71	0.39	55923	6667	35.7
			27	3116	523	2.44	24297	2658	14.37	866	71	0.39	50942	5545	29.63

Table 28. Continued

Design	MIP size			$K_{t,\bar{\lambda}+1}$			$K_{v-t,\bar{\lambda}+1}$			$K_{t,b-(\bar{\lambda}+1)}$			$K_{v-t,b-(\bar{\lambda}+1)}$		
t-(v, k, λ)	Col	Row	Nd	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec
2-(9,3,2)	28	2490	407	1.94	18762	2036	11.35	896	71	0.41	59812	7007	37.16		
			29	3107	571	2.58	20331	2324	12.69	837	71	0.38	46136	5279	29.04
			30	1978	289	1.38	21531	2158	11.89	822	71	0.37	48968	5437	29.12
			31	2575	419	1.95	14087	1558	8.46	844	71	0.37	45889	5038	27.85
			32	2512	445	2.1	18780	2078	11.16	877	71	0.39	54245	6535	34.65
			33	2555	427	1.98	24466	2586	14.03	836	71	0.37	53894	6312	32.91
			34	3834	743	3.27	18641	1999	10.76	891	71	0.38	72222	8419	44.46
			35	2577	427	2.01	17330	1938	10.45	879	71	0.38	62902	7286	38.9
			36	3157	585	2.7	15722	1820	9.77	907	71	0.38	48783	6035	31.84
2-(9,4,3)	189	488	1	38806	6111	23.32	7971	1243	4.65	810	71	0.3	20609	3027	12.09
			2	38523	5913	22.6	11786	1649	6.33	795	71	0.31	19274	2887	11.42
			3	38824	5961	22.44	8910	1295	5.04	813	71	0.32	18256	2839	11.47
			4	37582	5795	22.59	7555	1076	4.24	843	71	0.32	16942	2539	10.22
			5	37765	5887	22.84	7617	1116	4.33	749	71	0.3	21386	3215	12.81
			6	36447	5743	21.65	6666	1093	4.13	826	77	0.34	18564	2903	11.42
			7	38193	5981	22.78	10425	1437	5.63	768	71	0.31	21749	3293	13.2
			8	38148	6029	22.92	7418	1162	4.44	815	71	0.32	17734	2757	10.8
			9	38205	5927	22.25	8441	1237	4.79	795	71	0.3	23406	3497	13.95
			10	36247	5719	22.06	8566	1269	4.95	784	71	0.31	22996	3457	14.06
			11	37456	5803	22.54	7430	1079	4.22	772	71	0.32	21437	3325	13.07
2-(10,4,2)	175	452	1	9720	1815	6.22	5188	769	2.75	835	89	0.34	14532	2483	8.95
			2	9804	1777	6.15	5177	787	2.84	793	89	0.33	14917	2621	9.47
			3	9262	1653	5.65	6007	901	3.25	848	89	0.36	12517	2039	7.38
2-(11,5,2)	143	365	1	5367	1095	3.12	2738	494	1.41	889	109	0.33	3856	765	2.21
2-(13,3,1)	377	1016	1	225	12	0.12	44259	4155	38.14	1594	165	1.36	472	11	0.17
			2	231	12	0.12	46638	4139	38.19	1471	151	1.2	446	12	0.18
2-(13,4,1)	195	509	1	145	12	0.07	3874	683	2.9	1141	159	0.69	1166	61	0.4
2-(15,3,1)	575	1577	1	300	14	0.21	197156	14876	239.28	2839	205	2.94	626	12	0.34
			2	292	13	0.22	212787	17790	277.96	3054	231	3.27	962	28	0.62
			3	288	13	0.21	187872	14734	233.62	2954	205	3	837	23	0.53
			4	279	11	0.2	187339	14933	239.61	4273	341	4.92	597	20	0.42
2-(15,7,3)	255	677	1	64300	9643	48.62	8742	1421	6.97	2327	207	1.15	40114	5763	30.61
			2	63610	9567	48.82	11479	1835	9.09	2459	209	1.17	41708	5965	31.11
			3	64197	9485	49.08	9115	1523	7.39	2385	207	1.2	39924	5747	30.44
			4	62254	9325	46.98	8417	1439	6.95	2390	209	1.15	41535	5841	31.4
			5	63520	9461	47.99	9303	1583	7.85	2350	209	1.15	41538	6021	31.91
2-(16,4,1)	356	962	1	203	12	0.12	31971	3309	26.59	2458	237	1.86	540	16	0.2
2-(16,6,2)	288	770	1	16396	2715	15.53	16643	2179	13.05	2472	243	1.53	49896	7519	44.87
			2	15048	2473	13.81	18136	2479	14.74	2566	265	1.62	59719	9331	54.95
			3	18349	3043	16.87	16174	2087	12.6	2502	239	1.57	57997	8943	53.65
2-(19,9,4)	399	1085	1	514950	65017	563.45	67669	8451	71.21	4768	349	3.29	335818	36315	339.59
			2	516034	65953	561.46	59943	7603	64.77	4840	345	3.31	348821	38115	359.14
			3	512125	65285	556.69	65291	8335	71.94	5965	515	4.66	337776	36191	338.77
			4	511521	65535	559.14	64687	7954	68.63	4679	341	3.22	324742	34767	332
			5	521156	66391	576.12	64458	7965	68.8	6295	493	4.57	326500	35483	328.69
			6	520029	65781	578.24	74006	8859	77	4839	341	3.17	345347	38045	362.79
2-(25,4,1)	1325	3752	1	412	15	0.68	5823070	275493	18247.18	14874	675	33.2	1550	27	2.42

Table 29. Statistics on minimize WOBS for classes with parameter λ

Design	MIP size			$K_{t,\lambda+1}$			$K_{v-t,\lambda+1}$			$K_{t,b-(\lambda+1)}$			$K_{v-t,b-(\lambda+1)}$		
t-(v, k, λ)	Col	Row	Nd	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec
2 -(6,3,2)	76	182	1	279	51	0.07	614	113	0.16	817	116	0.18	336	62	0.09
2-(7,3,1)	63	149	1	42	3	0.00	471	81	0.1	403	82	0.11	98	7	0.02
2-(7,3,2)	119	296	1	91	5	0.01	3147	435	1.03	2107	237	0.61	195	13	0.04
			2	90	5	0.02	3127	433	1.03	2201	227	0.6	320	22	0.07
			3	91	5	0.02	2940	427	1	2014	212	0.55	201	14	0.04
			4	81	3	0.01	3298	425	1.05	1940	228	0.57	178	14	0.05
2-(7,3,3)	175	443	1	108	4	0.03	19694	3193	10.45	5117	399	1.6	231	15	0.08
			2	108	4	0.02	13380	1815	6.5	5017	380	1.58	242	17	0.09
			3	108	4	0.02	14968	2161	7.3	4887	399	1.59	228	14	0.08
			4	115	4	0.02	19624	3360	11.17	4981	405	1.61	230	14	0.07
			5	115	4	0.02	14275	2014	6.81	4999	363	1.52	249	16	0.07
			6	108	4	0.02	18620	3075	10.13	4909	399	1.59	238	17	0.08
			7	108	4	0.02	13084	1806	6.16	5189	381	1.59	228	14	0.08
			8	115	4	0.02	19280	3286	10.63	5072	386	1.62	239	6	0.05
			9	115	4	0.02	18291	3169	10.42	5106	416	1.63	240	16	0.08
			10	119	7	0.03	18862	3341	10.78	4933	384	1.59	248	22	0.11
2-(8,4,3)	134	338	1	563	59	0.16	7930	1179	3.17	1992	206	0.6	1047	176	0.45
			2	547	59	0.16	8363	1261	3.32	3723	336	1.04	1034	179	0.45
			3	688	107	0.29	7262	1043	2.79	1982	196	0.57	1106	198	0.49
			4	988	179	0.46	8444	1241	3.33	1982	203	0.58	918	160	0.41
2-(9,3,1)	129	326	1	49	4	0.01	1901	219	0.59	2066	235	0.64	212	6	0.03
2-(9,3,2)	249	650	1	107	3	0.02	16181	1561	7.84	5801	285	1.9	351	12	0.09
			2	107	3	0.02	17722	1621	8.41	5408	260	1.79	358	28	0.17
			3	107	3	0.02	21251	1964	10.11	5317	260	1.83	346	25	0.16
			4	107	3	0.02	18539	1787	9.02	5349	268	1.79	325	23	0.15
			5	107	3	0.03	19348	1855	9.46	5375	247	1.74	323	16	0.11
			6	107	3	0.02	16680	1599	8.1	5475	278	1.85	341	12	0.1
			7	107	3	0.02	20524	1960	10.06	5409	268	1.83	337	6	0.07
			8	107	3	0.02	16722	1613	8.19	5346	256	1.8	305	0	0.04
			9	107	3	0.02	17978	1777	9.19	5093	256	1.75	301	0	0.04
			10	107	3	0.02	16977	1678	8.34	5527	263	1.82	393	27	0.17
			11	107	3	0.03	24681	2151	11.35	5542	273	1.89	318	8	0.08
			12	107	3	0.02	17423	1709	8.68	5321	264	1.79	309	0	0.04
			13	118	4	0.03	17253	1615	8.1	5231	253	1.78	449	35	0.21
			14	118	4	0.02	16546	1554	7.79	5374	252	1.75	408	32	0.2
			15	118	4	0.03	17503	1643	8.39	5223	252	1.72	401	30	0.18
			16	118	4	0.03	17758	1716	8.55	5241	249	1.71	497	38	0.23
			17	118	4	0.03	23907	2227	11.68	5141	291	1.86	306	0	0.05
			18	118	4	0.03	12261	1133	5.92	5451	294	1.97	380	5	0.08
			19	107	3	0.02	23920	2192	11.79	5479	269	1.87	338	11	0.09
			20	107	3	0.04	16687	1563	8.23	5236	248	1.79	431	30	0.18
			21	107	3	0.03	18787	1832	9.45	5295	257	1.77	357	9	0.09
			22	107	3	0.02	17271	1606	8.25	5489	246	1.8	351	8	0.08
			23	107	3	0.03	24720	2353	12.41	5088	258	1.75	393	26	0.17
			24	107	3	0.02	17234	1638	8.53	5432	268	1.86	601	53	0.32
			25	107	3	0.02	16355	1599	8.26	5366	260	1.8	367	22	0.14
			26	107	3	0.03	23205	2133	11.16	5225	286	1.89	368	29	0.17
			27	107	3	0.02	17545	1674	8.75	5364	256	1.82	351	17	0.12

Table 29. Continued

Design	MIP size			$K_{t,\lambda+1}$			$K_{v-t,\lambda+1}$			$K_{t,b-(\lambda+1)}$			$K_{v-t,b-(\lambda+1)}$		
t-(v, k, λ)	Col	Row	Nd	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec
2-(9,3,2)	28	107	3	0.02	17328	1667	8.56	5467	242	1.81	376	13	0.11		
			29	107	3	0.02	12424	1329	6.74	5136	241	1.75	365	26	0.16
			30	118	4	0.03	22289	2089	10.97	5324	259	1.78	468	37	0.21
			31	118	4	0.03	18206	1749	8.93	5545	288	1.9	311	0	0.05
			32	118	4	0.03	24572	2227	12.01	5501	308	2.11	319	0	0.05
			33	118	4	0.02	18471	1925	9.69	5188	263	1.74	390	27	0.17
			34	118	4	0.03	17459	1635	8.31	5507	294	1.98	389	16	0.12
			35	118	4	0.02	24353	2215	11.49	5023	258	1.74	360	28	0.17
			36	118	4	0.03	17009	1609	8.21	5337	291	1.93	372	10	0.1
2-(9,4,3)	189	488	1	107	4	0.02	28242	3101	12.08	7970	481	2.34	288	19	0.1
			2	107	4	0.02	26820	2813	11.34	8009	497	2.39	296	16	0.09
			3	107	4	0.02	24796	2939	11.44	7683	495	2.42	297	16	0.09
			4	107	4	0.02	26819	3275	12.46	8278	539	2.57	354	21	0.12
			5	107	4	0.02	26245	3007	11.96	8352	517	2.52	486	44	0.21
			6	99	6	0.03	27773	3159	12.1	7914	519	2.43	298	24	0.11
			7	107	4	0.02	21895	2415	9.28	7548	505	2.37	283	15	0.09
			8	107	4	0.02	26919	2941	11.65	8088	516	2.47	285	13	0.08
			9	107	4	0.02	28501	3035	12.08	7886	489	2.35	335	21	0.11
			10	107	4	0.02	27483	2947	11.52	8158	551	2.57	346	24	0.13
			11	107	4	0.03	26462	2855	11.55	7937	503	2.43	365	29	0.14
2-(10,4,2)	175	452	1	114	9	0.04	11174	1207	4.44	5179	387	1.62	263	19	0.09
			2	114	9	0.04	11415	1265	4.69	5939	422	1.86	302	23	0.1
			3	126	9	0.04	10730	1133	4.36	5931	444	1.93	290	22	0.1
2-(11,5,2)	143	365	1	75	4	0.01	5434	685	2.05	2328	261	0.82	307	25	0.09
2-(13,3,1)	377	1016	1	95	3	0.04	22089	1581	13.83	12697	434	5.65	477	5	0.15
			2	95	3	0.03	22122	1601	13.88	13009	449	5.84	513	33	0.34
2-(13,4,1)	195	509	1	77	7	0.03	7197	563	2.8	6540	590	2.81	259	14	0.09
2-(15,3,1)	575	1577	1	148	5	0.09	55734	3175	46.76	36483	867	21.94	779	4	0.38
			2	148	5	0.1	54511	3140	45.79	34959	853	21.09	729	21	0.56
			3	148	5	0.09	56648	3194	47.88	35056	837	21.31	790	23	0.59
			4	148	5	0.08	54056	3152	46.79	34975	832	21.34	770	0	0.33
2-(15,7,3)	255	677	1	113	4	0.02	53111	4793	26.29	12623	908	5.18	418	30	0.19
			2	111	4	0.03	56025	4797	26.79	9412	678	4.01	348	18	0.14
			3	113	4	0.02	47948	4554	24.73	14926	1115	6.47	422	29	0.19
			4	113	4	0.03	56070	4983	27.84	12424	869	5.11	576	43	0.26
			5	114	4	0.03	57113	4873	27.02	13429	1006	5.73	343	18	0.13
2-(16,4,1)	356	962	1	64	4	0.03	18439	749	7.97	29006	1237	12.96	500	7	0.17
2-(16,6,2)	288	770	1	86	3	0.02	49430	3323	22.82	11970	721	5.07	426	22	0.19
			2	86	3	0.02	47094	3055	21.17	11997	723	5.1	407	25	0.2
			3	86	3	0.03	45106	3137	21.71	13150	789	5.64	390	23	0.19
2-(19,9,4)	399	1085	1	109	7	0.06	433260	30222	292.99	34883	1745	18.01	1107	64	0.73
			2	109	7	0.06	419946	30023	288.9	48766	2491	26.35	827	41	0.52
			3	109	7	0.06	411863	28421	277.83	34460	1807	18.68	866	54	0.6
			4	109	7	0.06	427837	28568	282.45	34909	1817	19	795	40	0.51
			5	109	7	0.06	424676	28557	283.91	33610	1658	17.67	701	35	0.44
			6	109	7	0.07	437331	31227	301.7	33139	1764	18.05	676	30	0.4
2-(25,4,1)	1325	3752	1	245	49	1.49	640142	16777	1022.92	654064	9237	1140.58	1955	0	2.52

Table 30. Statistics on minimize WOBS for other classes with parameter $\overline{\lambda}$

Design	MIP size			$K_{t,\overline{\lambda}+1}$			$K_{v-t,\overline{\lambda}+1}$			$K_{t,b-(\overline{\lambda}+1)}$			$K_{v-t,b-(\overline{\lambda}+1)}$		
t-(v, k, λ)	Col	Row	Nd	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec
2-(6,3,2)	76	182	1	279	51	0.07	614	113	0.18	817	116	0.18	336	62	0.1
2-(7,3,1)	63	149	1	231	45	0.06	432	103	0.13	331	69	0.08	264	57	0.07
2-(7,3,2)	119	296	1	1039	157	0.36	6791	1457	3.16	1221	211	0.47	817	125	0.3
			2	1392	235	0.52	6418	1299	2.72	1118	197	0.43	1143	175	0.42
			3	1109	171	0.38	7570	1592	3.46	1119	183	0.41	853	143	0.32
			4	1119	173	0.38	5038	1073	2.36	1200	233	0.49	892	168	0.39
2-(7,3,3)	175	443	1	1145	143	0.5	78417	15143	47.21	3078	502	1.55	1945	286	1.07
			2	882	87	0.32	117057	21175	66.54	2984	589	1.76	1893	307	1.19
			3	1174	145	0.49	117064	22915	71.94	10106	1569	5.11	1903	277	0.99
			4	1332	169	0.67	118531	20521	66.03	3589	605	1.95	1760	244	0.9
			5	1038	133	0.47	98918	17451	55.88	8878	1542	4.84	2083	298	1.04
			6	1237	151	0.52	91923	17675	54.97	3230	624	1.85	1985	338	1.15
			7	948	95	0.33	101119	16725	53.29	8466	995	3.49	1906	285	1.02
			8	1503	255	0.81	112414	20497	68.88	6899	1230	3.96	1819	245	0.86
			9	1176	201	0.73	101517	19805	65.3	5815	1197	3.59	1838	245	0.88
			10	1153	153	0.53	66370	12503	39.55	8712	1140	3.95	1782	243	0.83
2-(8,4,3)	134	338	1	563	59	0.17	7930	1179	3.17	1992	206	0.59	1047	176	0.46
			2	547	59	0.17	8363	1261	3.42	3723	336	1.05	1034	179	0.46
			3	688	107	0.28	7262	1043	2.84	1982	196	0.57	1106	198	0.52
			4	988	179	0.44	8444	1241	3.39	1982	203	0.58	918	160	0.4
2-(9,3,1)	129	326	1	771	101	0.25	1195	196	0.48	771	101	0.25	1195	196	0.49
2-(9,3,2)	249	650	1	4000	355	1.85	102657	12339	61.34	8296	845	4.08	7135	872	4.29
			2	5479	507	2.63	109075	14294	71.05	2007	247	1.15	6276	823	4.17
			3	3722	353	1.79	107743	13769	67.66	2387	245	1.19	8704	922	4.84
			4	5200	483	2.49	93160	11988	58.51	2274	229	1.13	6440	807	4.01
			5	3662	357	1.75	114025	15372	72.02	2312	321	1.49	7831	1048	5.14
			6	4406	441	2.21	89812	11462	56.59	5504	699	3.29	5386	706	3.52
			7	5153	521	2.57	96108	12146	59.99	2376	341	1.55	7873	979	4.99
			8	3073	261	1.39	121541	15457	77.2	2498	261	1.27	7009	971	4.82
			9	4330	419	2.09	101481	12807	63.88	3060	379	1.81	11766	1482	7.37
			10	3662	321	1.67	108266	13843	68.1	2072	251	1.16	6831	821	4.23
			11	5704	539	2.76	109487	13934	68.92	2388	359	1.61	7969	1075	5.27
			12	3955	339	1.76	94672	12019	60.37	4353	500	2.4	7416	1027	4.92
			13	5577	549	2.82	108353	13892	68.41	1969	237	1.13	8407	1074	5.33
			14	5027	482	2.48	92403	12077	58.96	6498	1031	4.58	8737	1207	5.95
			15	5654	529	2.67	92399	11570	57.77	4062	547	2.49	7715	993	5.01
			16	5661	561	2.81	105185	13915	68.24	2229	247	1.21	8304	1117	5.46
			17	3471	315	1.61	106592	13821	68.53	2038	253	1.18	7869	986	4.95
			18	4212	443	2.24	115338	14686	74.6	1973	269	1.25	13565	1677	8.61
			19	3198	289	1.49	111601	14120	71	2074	233	1.12	5873	734	3.73
			20	5702	581	2.93	101411	12812	64.44	1852	209	1.01	6384	776	3.93
			21	6218	637	3.2	125018	15877	79.16	2014	227	1.09	8436	1095	5.59
			22	4611	493	2.43	111424	14238	69.85	6015	733	3.41	7932	1085	5.42
			23	5657	531	2.73	113306	15221	75.62	2368	261	1.28	5731	673	3.55
			24	4652	475	2.44	90321	11736	58.36	2823	411	1.91	6624	831	4.21
			25	3439	325	1.69	90872	11781	59.04	3000	377	1.85	6345	781	4.05
			26	4989	517	2.57	111455	14013	70.7	1915	209	1.05	6283	817	4.19
			27	3812	391	1.96	102306	13318	65.06	2943	387	1.76	8167	1084	5.36

Table 30. Continued

Design	MIP size			$K_{t,\bar{\lambda}+1}$			$K_{v-t,\bar{\lambda}+1}$			$K_{t,b-(\bar{\lambda}+1)}$			$K_{v-t,b-(\bar{\lambda}+1)}$		
t-(v, k, λ)	Col	Row	Nd	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec	ite	bbn	sec
2(9,3,2)	249	650	28	2601	239	1.25	82565	9832	50.34	2285	356	1.56	9100	1242	6.22
			29	5047	511	2.58	102522	14244	73	2653	379	1.74	10007	1148	6.17
			30	6888	673	3.45	97508	13298	66.69	3975	475	2.22	9106	1257	6.15
			31	5333	509	2.64	78020	9384	47.72	2282	285	1.31	7130	869	4.59
			32	5672	533	2.81	81596	10028	49.74	1966	231	1.13	7972	989	4.94
			33	6001	639	3.12	112155	14020	68.58	2341	309	1.43	8027	1050	5.27
			34	5529	515	2.62	89581	10694	53.56	2274	251	1.19	9206	1213	5.95
			35	2550	233	1.21	110595	14207	71.31	2048	233	1.12	8969	1216	6.03
			36	2872	267	1.33	112974	14256	70.88	2284	251	1.2	8209	1017	5.11
2-(9,4,3)	189	488	1	1129	133	0.47	72405	10573	37.96	9033	1188	4.29	2976	490	1.78
			2	1094	123	0.44	48681	7463	26.5	4478	423	1.72	2477	408	1.5
			3	1787	265	0.93	69048	11177	39.49	4875	525	2.07	2266	358	1.34
			4	1611	231	0.84	71685	10505	38.48	4476	424	1.79	2614	472	1.71
			5	1541	209	0.76	77621	11067	40.43	4532	441	1.73	2657	440	1.59
			6	1550	223	0.77	75080	11211	39.89	9386	1345	4.58	2500	468	1.64
			7	1231	145	0.55	62595	9635	34.13	8119	1183	4.06	2439	376	1.4
			8	1071	89	0.37	76155	10985	40.89	4665	468	1.84	2924	497	1.79
			9	1226	151	0.53	76240	11405	40.78	4560	440	1.77	2294	349	1.29
			10	1053	105	0.41	70541	10207	38.32	4750	487	1.92	2975	473	1.75
			11	1661	249	0.88	78388	11753	42.58	4441	449	1.82	2080	341	1.24
2-(10,4,2)	175	452	1	1782	253	0.87	19852	3535	11.67	2408	252	0.92	2373	397	1.36
			2	1770	233	0.81	18559	3323	11.04	2361	240	0.9	2649	395	1.4
			3	1557	197	0.67	13252	2463	8.18	4155	499	1.74	2349	374	1.34
2-(11,5,2)	143	365	1	1167	213	0.53	6000	965	2.68	2875	397	1.09	1286	238	0.65
2-(13,3,1)	377	1016	1	16487	891	8.82	530	11	0.21	142	5	0.05	27763	1979	18.46
			2	17952	948	9.45	611	22	0.3	142	5	0.06	40454	3061	27.69
2-(13,4,1)	195	509	1	2262	225	1	302	14	0.09	125	6	0.03	2863	398	1.64
2-(15,3,1)	575	1577	1	58303	1831	35.66	846	11	0.48	190	8	0.12	104492	4679	81.41
			2	64920	2100	40.11	819	26	0.65	190	8	0.12	108575	4905	85.52
			3	67134	2194	41.95	971	19	0.63	190	8	0.13	107797	4813	84.37
			4	63584	2103	40.21	864	11	0.51	190	8	0.12	111897	5035	87.22
2-(15,7,3)	255	677	1	4437	625	2.88	81839	8539	45.47	11755	984	5.28	4560	750	3.65
			2	5319	831	3.77	77428	8357	44.27	12345	1061	5.6	4753	734	3.59
			3	4781	715	3.18	73019	8107	42.37	14519	1339	7.03	4703	770	3.82
			3	4781	715	3.18	73019	8107	42.37	14519	1339	7.03	4703	770	3.82
			4	4325	665	3.01	72564	8095	41.99	10959	933	4.96	4636	727	3.56
			5	5418	825	3.73	70575	7641	40.73	12902	1152	6.16	5111	763	3.79
2-(16,4,1)	356	962	1	6237	321	2.89	494	5	0.15	129	5	0.05	38855	2923	24.7
2-(16,6,2)	288	770	1	6160	691	3.94	61622	7310	42.58	8935	947	5.36	9344	1094	6.51
			2	6463	753	4.16	61005	6957	41.11	6422	587	3.58	8618	1113	6.52
			3	5048	503	2.86	60588	7041	42.17	6610	614	3.71	8096	1033	6.24
2-(19,9,4)	399	1085	1	8857	1223	9.36	532182	37571	357.99	49015	3461	30.83	12397	1561	13.41
			2	9144	1287	9.69	645940	51886	475.25	38945	2265	22.16	18032	2089	18.55
			3	7650	983	7.45	620907	48455	454.51	50980	3709	33.28	13642	1795	15.27
			4	8961	1159	8.89	624841	48832	453.01	50033	3484	31.88	20235	2201	19.49
			5	8849	1221	9.09	696780	55465	521.87	101605	7595	69.32	13925	1766	15.45
			6	8136	1125	8.28	690259	53103	497.4	34432	2125	20.12	13425	1698	14.74
2-(25,4,1)	1325	3752	1	872710	16535	1267.33	2063	57	4.6	417	28	1.15	8785788	214182	18756.52

VIII.3. Cutting-plane algorithm on substructure cuts

Results of the implementation of the cutting-plane algorithm presented in Section VIII.3 are given in Table 31. The headings for the tables are as follows: *NCols* is the number of variables in the formulation. *NRini* is number of rows in the base formulation. *NRend* is the number of rows in the LP model when the solution was found. *SIt* is the number of simplex iterations. *Its* is the number of iterations that the cutting-plane algorithm required to find the solution. *BBn* is the number of branch-and-bound nodes evaluated, which in this case is zero for all instances. *Sec* is the CPU time in seconds in a PC Pentium III, 664 MHz, 256 MB of RAM. The test instances are from Mathon and Rosa [40] and from Kreher [37].

From the results in Table 31, we conclude that this cutting-plane algorithm is very fast, and is capable of obtaining the solution at the root node of the search tree. It is worth to note that the added cuts in this cutting-plane algorithm are local, in contrast with the biclique cuts that are utilized in a branch-and-cut algorithm described in the next section.

VIII.4. Branch-and-cut algorithm

We implemented a branch-and-cut algorithm on our new formulation GDP. We consider three factors that can be varied, and test instances of block designs on them. The instances are from Mathon and Rosa [40].

The first factor we considered is different base formulations to start the branch-and-cut algorithm. A *base formulation* is the set of inequalities to be added a priori. The most natural base formulation is to start with the "easy" star constraints and then successively add the "hard" biclique constraints until a design is found. We test this option and also the minimal star inequalities and the class $K_{v-1,b-1}$ to measure the effect of the integrality property on the search results. Other factor we studied is the effect of a cold start, with respect to fix one point or one block a priori, which we call warm start. The third factor is using the original biclique inequalities or equivalent classes.

Table 31. Results for cutting-plane algorithm using substructure cuts

t	v	k	λ	b	r	NCols	NRini	NRend	SIt	Its	BBn	Sec
2	4	3	2	4	3	16	8	14	17	0	0	0.01
2	6	3	2	10	5	60	16	46	70	2	0	0.01
2	7	3	1	7	3	49	14	38	51	1	0	0.00
2	7	3	2	14	6	98	21	69	114	3	0	0.01
2	7	3	3	21	9	147	28	76	182	3	0	0.02
2	8	4	3	14	7	112	22	78	169	3	0	0.06
2	9	3	1	12	4	108	21	85	104	3	0	0.01
2	9	3	2	24	8	216	33	129	301	5	0	0.04
2	13	4	1	13	4	169	26	218	266	7	0	0.05
3	8	4	1	14	7	112	22	190	151	6	0	0.03
3	10	4	1	30	12	300	40	552	664	15	0	2.81
3	5	4	2	5	4	25	10	22	27	0	0	0.00
3	8	4	2	28	14	224	36	132	295	3	0	0.04
3	6	4	3	15	10	90	21	53	100	1	0	0.01
3	8	4	3	42	21	336	50	266	581	8	0	0.11

We use a bit mask to specify the levels of the factors mentioned above. This bit mask concept is taken from OSL [32]. A *bit mask* is a bit string that is used to specify one or more options by using each bit as an ″on-off″ switch. The levels of the factor to be used used will be the sum of the numbers represented by the individual bits. For a given decimal bitmask value, the options will be retrieved by using the binary representation of that decimal number. Table 32 specifies the bits and values for the different levels of the factors.

One implementation detail is the following. Since the value of the objective function of GDP is known for all t-designs, that is $vr = bk$, then we can set the cut-off parameter in the search tree to discard branches of the tree is the solution at a particular node is worse than $vr = bk$ with certain tolerance.

VIII.4.1. Computational results

Some results of the branch-and-cut application are shown in Tables 33, 34, 35, 36, 37, 38, and 39. The bit mask values used are listed in the order: base formulation bit mask, warm start formulation bit mask, and separation problem bit mask. Those are abbreviated as {B,W,S} in the title of each table. The headings for the tables are as follows: *NCols* is the number of variables in the formulation. *NRini* is number of rows in the base formulation. *NRend* is the number of rows at the branch-and-bound node where the solution was found. *SIt* is the number of simplex iterations. *BBn* is the number of branch-and-bound nodes evaluated. *Sec* is the time in seconds. A note “nodes” under the column *NRend* indicates that the algorithm failed to find a solution after 50,000 nodes.

We observed that there is no clear winning combination of levels for the factors, since some instances solved in fewer nodes when using warm start, but others performed better with the cold start. The base formulation that includes both point and block star inequalities, as well as substructure inequalities, seems to be the most promising combination in terms of the number of instances successfully solved.

Table 32. Factor levels as bit values

Factor	Bit	Bitmask value	Factor level
Warm start	0	1	Fix one block
	1	2	Fix one point
	2	4	Use Tonchev lower bound (2.6)
	3	8	Do not use warm start
Base formulation	0	1	star $K_{1,b}$ inequalities
	1	2	star $K_{v,1}$ inequalities
	2	4	derived design inequalities
	3	8	residual design inequalities
	4	16	dual derived inequalities
	5	32	dual residual inequalities
	6	64	minimal point-star inequalities $K_{v-1,1}$
	7	128	minimal block-star inequalities $K_{1,b-1}$
	8	256	class $K_{v-1,b-1}$
Separation Class	0	1	Class $K_{t,\lambda+1}$
	1	2	Class $K_{v-t,\lambda+1}$
	2	4	Class $K_{t,b-(\lambda+1)}$
	3	8	Class $K_{v-t,b-(\lambda+1)}$
	4	16	Class $K_{t,\bar{\lambda}+1}$
	5	32	Class $K_{v-t,\bar{\lambda}+1}$
	6	64	Class $K_{t,b-(\bar{\lambda}+1)}$
	7	128	Class $K_{v-t,b-(\bar{\lambda}+1)}$

Table 33. Branch-and-cut run for bit mask values {B,W,S} = {3,8,1}

t	v	k	λ	b	r	NCols	NRini	NRend	SIt	BBn	Sec
2	4	3	2	4	3	16	8	8	21	0	0.13
2	6	3	2	10	5	60	16	25	1112	20	1.73
2	7	3	1	7	3	49	14	17	2163	68	5.23
2	7	3	2	14	6	98	21	37	4181	58	4.87
2	7	3	3	21	9	147	28	56	11392	133	20.35
2	8	4	3	14	7	112	22	36	137901	1913	143.35
2	9	3	1	12	4	108	21	42	4515	65	5.02
2	10	4	2	15	6	150	25	65	169002	1442	107.33
2	11	5	2	11	5	121	22	84	32235	263	20.57
2	13	3	1	26	6	338	39	237	1017665	5122	167.61
2	13	4	1	13	4	169	26	36	1210	10	0.89
2	16	4	1	20	5	320	36	nodes	14953730	50000	3064.17
2	16	6	2	16	6	256	32	nodes	17785230	50000	3939.82

Table 34. Branch-and-cut run for bit mask values {B,W,S} = {3,3,1}

t	v	k	λ	b	r	NCols	NRini	NRend	SIt	BBn	Sec
2	4	3	2	4	3	16	8	8	14	0	0.14
2	6	3	2	10	5	60	16	25	1637	44	3.27
2	7	3	1	7	3	49	14	23	194	6	0.65
2	7	3	2	14	6	98	21	33	2067	38	3.38
2	7	3	3	21	9	147	28	51	822458	10787	397.45
2	8	4	3	14	7	112	22	34	149854	1821	128.57
2	9	3	1	12	4	108	21	58	39511	558	40.37
2	10	4	2	15	6	150	25	89	44125	372	28.20
2	11	5	2	11	5	121	22	63	6304	48	4.55
2	13	3	1	26	6	338	39	93	845778	4561	189
2	13	4	1	13	4	169	26	119	96753	799	53.19
2	16	4	1	20	5	320	36	nodes	12735712	50000	2882.30
2	16	6	2	16	6	256	32	nodes	15317726	50000	3300.03

Table 35. Branch-and-cut run for bit mask values {B,W,S} = {3,4,1}

t	v	k	λ	b	r	NCols	NRini	NRend	SIt	BBn	Sec
2	4	3	2	4	3	16	8	8	16	0	0.15
2	6	3	2	10	5	60	16	19	233	8	0.65
2	7	3	1	7	3	49	14	22	775	24	1.95
2	7	3	2	14	6	98	21	35	2822	51	3.77
2	7	3	3	21	9	147	28	106	487837	8341	913.70
2	8	4	3	14	7	112	22	38	11313	148	12.24
2	9	3	1	12	4	108	21	51	69029	939	56.70
2	10	4	2	15	6	150	25	88	1227658	9979	796.38
2	11	5	2	11	5	121	22	119	168511	1529	108.64
2	13	3	1	26	6	338	39	nodes	9945544	50000	2855.07
2	13	4	1	13	4	169	26	nodes	7711981	50000	1874.52
2	16	4	1	20	5	320	36	nodes	9531077	50000	1766.85
2	16	6	2	16	6	256	32	nodes	17569487	50000	4504.75

Table 36. Branch-and-cut run for bit mask values {B,W,S} = {15,3,1}

t	v	k	λ	b	r	NCols	NRini	NRend	SIt	BBn	Sec
2	4	3	2	4	3	16	14	14	11	0	0.14
2	6	3	2	10	5	60	26	31	178	5	0.58
2	7	3	1	7	3	49	26	38	67	1	0.28
2	7	3	2	14	6	98	33	39	1509	28	2.40
2	7	3	3	21	9	147	40	47	5121	64	10.10
2	8	4	3	14	7	112	36	46	6726	97	8.46
2	9	3	1	12	4	108	37	62	1930	36	2.81
2	10	4	2	15	6	150	43	121	43684	437	32.83
2	11	5	2	11	5	121	42	78	23743	207	15.68
2	13	3	1	26	6	338	63	121	324610	2170	173.63
2	13	4	1	13	4	169	50	88	7649	65	5.50
2	16	4	1	20	5	320	66	nodes	12770397	50000	3271.10
2	16	6	2	16	6	256	62	194	2647963	8027	1550.66

Table 37. Branch-and-cut run for bit mask values {B,W,S} = {31,3,1}

t	v	k	λ	b	r	NCols	NRini	NRend	SIt	BBn	Sec
2	4	3	2	4	3	16	17	17	11	0	0.14
2	6	3	2	10	5	60	35	37	189	6	0.61
2	7	3	1	7	3	49	32	39	173	6	0.65
2	7	3	2	14	6	98	46	nodes	1916805	50000	1273.81
2	9	3	1	12	4	108	48	64	5239	94	7.29
2	10	4	2	15	6	150	57	72	142412	1678	112.15
2	11	5	2	11	5	121	52	68	6538	60	5.07
2	13	3	1	26	6	338	88	189	264948	1512	128.64
2	13	4	1	13	4	169	62	106	15541	149	11.05
2	16	4	1	20	5	320	85	295	744776	2987	302.61
2	16	6	2	16	6	256	77	126	1943614	5916	863.30

Table 38. Branch-and-cut run for bit mask values {B,W,S} = {39,3,1}

t	v	k	λ	b	r	NCols	NRini	NRend	SIt	BBn	Sec
2	4	3	2	4	3	16	14	14	10	0	0.14
2	6	3	2	10	5	60	30	33	148	3	0.44
2	7	3	1	7	3	49	26	40	69	1	0.27
2	7	3	2	14	6	98	40	nodes	2137009	50000	1295.02
2	9	3	1	12	4	108	40	44	1238	29	2.25
2	10	4	2	15	6	150	48	62	9584	87	7.67
2	11	5	2	11	5	121	42	83	4411	35	3.29
2	13	3	1	26	6	338	76	220	617589	3747	317.35
2	13	4	1	13	4	169	50	94	6136	53	4.66
2	16	4	1	20	5	320	70	214	2147629	9032	933.64
2	16	6	2	16	6	256	62	346	1007841	3169	493.43

Table 39. Branch-and-cut run for bit mask values {B,W,S} = {63,3,1}

t	v	k	λ	b	r	NCols	NRini	NRend	SIt	BBn	Sec
2	4	3	2	4	3	16	20	20	11	0	0.14
2	6	3	2	10	5	60	44	48	88	2	0.36
2	7	3	1	7	3	49	38	41	118	4	0.50
2	7	3	2	14	6	98	59	nodes	1890382	50000	1270.50
2	7	3	3	21	9	147	80	nodes	3153870	50000	2061.21
2	9	3	1	12	4	108	59	64	1521	34	2.64
2	10	4	2	15	6	150	71	93	10081	89	7.72
2	11	5	2	11	5	121	62	86	5314	55	4.71
2	13	3	1	26	6	338	113	187	3757285	22898	932.02
2	13	4	1	13	4	169	74	132	14849	145	11.08
2	16	4	1	20	5	320	104	nodes	13212752	50000	3374.48
2	16	6	2	16	6	256	92	556	5027869	10849	692.13

CHAPTER IX

CONCLUSIONS AND FUTURE WORK

This dissertation gives a new promising approach to combinatorial t-designs that is useful in constructing t-designs by polyhedral methods. Our approach starts with a new problem equivalence result, that leads to a novel integer programming description, and results in some strong classes of valid inequalities. We provide several theoretical results, and show their validity and usefulness in a computational study.

The combinatorial t-design problem has several applications, mainly in computer science, statistics and communications. Also, special cases of t-designs represent several other combinatorial structures like affine planes and geometries, projective planes and geometries, Steiner systems, Hadamard designs, among others. Any of those examples of problems alone, may have been the subject of ample research by itself. This highlights the broadness of the scope and possible applicability of our study. Moreover, our work also contributes with new results to join the only few polyhedral studies that have been conducted for t-designs.

One of the supporting arguments for the relevance of the research question that drove this dissertation is the fact that existence results for t-designs are far from settled in general. There are many open existence problems for block designs and t-designs. Most effort to date in establishing existence results has been invested basically in triple systems 2-($v, 3, \lambda$), quadruple systems 3-($v, 4, \lambda$), and Steiner systems t-($v, k, 1$).

One of the main contributions of our work is to show that the problem of finding a t-design has an equivalent graph problem, which is the biclique-free b-

factor problem in bipartite graphs. This graph problem also has been studied only for some small-parameter particular instances, not in general. Examples are the triangle-free 2-factor, and the square-free 2-factor. We believe that this equivalence result opens the possibilities of further research, like for example, may lead to a proof for the complexity of the problem that, to the best of our knowledge, has not been reported in the literature. The new polyhedral description GDP that we propose in this work goes therefore beyond an only application of an existent integer programming model.

Compared with the basically two existent integer programming formulations for t-designs, an advantage of GDP over the set-covering or set-packing model is that it does not have a combinatorial number of variables. Also, GDP has the advantage over the other existent integer programming of not having any nonlinear inequalities. On the other hand, a disadvantage is that the number of rows in our formulation GDP is very large, but it is suitable for a branch-and-cut, as the computational results show.

Other contribution of our work is the polyhedral analysis including the result on the dimension of the polyhedron associated with GDP. The dimension is an important aspect, since many results in polyhedral theory are applied easier (or sometimes exclusively) to full-dimensional polyhedra. Also, we present an analysis of the other two polyhedra involved in our formulation.

Results in linear programming and theory of valid inequalities were also linked to t-designs in this work. The converse is also true, since theorems from design theory, definitions, and certain operations on the incidence structure of a t-design were also successfully linked to obtain new polyhedral results. The contribution in this aspect was to derive strong classes of valid inequalities for GDP, and to show that some of them are implied equations. The stable-set class of valid inequalities is an alternative problem equivalence for t-designs, that is, a t-design is a restricted b-factor in bipartite graph where the restiction is being stable-set free instead of biclique-free.

The new classes of valid inequalites, used as cuts in a branch-and-cut algorithm, proved efficient to reduce the search tree in the computational tests. Also, the substructure cuts used in a cutting-plane algorithm proved capable of obtaining a t-design at the root node of the search tree. We also proved analytically that other classes of valid inequalities can be used in lieu of the original biclique inequalities of GDP, but that not all of them give the same linear programming bounds.

Other contribution of our work was to derive an integer programming model for the separation problem. Due to the combinatorial nature of the biclique inequalities, the number of them rapidly increases as the size of the instance increases. By solving an optimal weighted biclique subgraph problem, we can avoid going through the entire class to verify the unsatisfied biclique inequalities. Although the separation problem is difficult by itself, it turned out to be solved consistently fast for all our test instances.

Until now, one cannot speak of *the* polyhedron associated with seemingly difficult problems like t-designs. When a complete description by means of linear inequalities is not available, most likely there will be several formulations for a particular problem. The choice of them will be crucial for the success of implicit enumeration techniques like branch-and-cut. This work proposed and studied GDP to examine if it has favorable characteristics for the application of this technique.

It is out hope that the reader would find the approach and results presented in this work motivating. We now present some future research directions.

As mentioned before, GDP can be used to find packing an covering designs.

The drawback for these problems is that the unknown parameter will be the number of blocks. Schönheim bounds may provide a good starting value for this parameter.

In our work, no post-optimality procedure was done to find further nonisomorphic solutions. This would imply searching for alternative optimal solutions for the optimization problem, and then test for isomorphism among them.

Other research direction would be the application of integer programming Lagrangian Relaxation. In general, this method applies when we have an IP with some "nice" constraints, and other constraints much more difficult to solve, that can be called "hard". Then, instead of explicitly enforcing the hard constraints, we put then into the objective function with some penalty coefficients. This relaxed problem solution provides a bound for the optimal, and can be used to solve the original problem.

Regarding the graph approach, recently the topic of perfect graphs has received a lot of attention. These type of graphs have some nice properties. We suggest that there may be an interesting link of perfect graphs to the biclique-free b-factor studied in this work, since it has been proven that any bipartite graph (an also the complement of a bipartite graph) is perfect.

Algorithmically, a proposed research direction is to investigate more efficient techniques to solve the separation problem. In our research, we formulated the separation problem as a MIP, and used simplex-based branch-and-bound to solve it. Since in practice it is common that the separation problem does not need to be solved exactly, usually approximate solution suffices. Therefore it is worth investigating a heuristic that can be used instead of the MIP to find violated inequalities to add at each node of the search tree.

In our search for literature related to combinatorial designs, we found that some economic theory problems may be related. Those are problems in cooperative game theory, like financial cooperative games, bankruptcy games, voting situations, winning coalitions, bargaining set. One property that distinguish the optimality of these cooperative games is to be totally balanced. We believe that the relation to the balanced property of some combinatorial designs is worth investigation.

REFERENCES

[1] R. J. R. ABEL and S. C. FURINO, 1996. Resolvable and near resolvable designs, in *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 87–94.

[2] R. J. R. ABEL and M. GREIG, 1996. BIBDs with small block size, in *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 41–47.

[3] R. J. R. ABEL and M. GREIG, 1997. Some new RBIBDs with block size 5 and PBDs with block sizes $\equiv 1 \bmod 5$, *Australasian Journal of Combinatorics 15*, 177–202.

[4] I. ANDERSON, 1997. *Combinatorial Designs and Tournaments*, Oxford Lecture Series in Mathematics and its Applications 6, Oxford University Press, Oxford, United Kingdom.

[5] G. AUSIELLO, P. CRESCENZI, G. GAMBOSI, V. KANN, A. MARCHETTI-SPACCAMELA, and M. PROTASI, 1999. *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*, Springer-Verlag, Berlin, Germany.

[6] A. BACHEM and W. KERN, 1992. *Linear Programming Duality: An Introduction to Oriented Matroids*, Springer-Verlag, Berlin, Germany.

[7] S. Y. BERKOVICH, 1994. Multiprocessor interconnection network using pairwise balanced combinatorial designs, *Information Processing Letters 50*, 217–222.

[8] T. BETH, D. JUNGNICKEL, and H. LENZ, 1999. *Design Theory*, Volume I,II

of Encyclopedia of Mathematics and its Applications, Cambridge University Press, Cambridge, United Kingdom, 2nd ed.

[9] R. E. BIXBY, S. CERIA, C. MCZEAL, and M. W. SAVELSBERGH, 1998. An updated mixed integer programming libary: MIPLIB 3.0, *Optima 58*, 12–15.

[10] I. BLUSKOV and K. HEINRICH, 1999. General upper bounds on the minimum size of covering designs, *Journal of Combinatorial Theory, Series A 86*, 205–213.

[11] P. J. CAMERON, 1994. *Combinatorics: Topics, Techniques, Algorithms*, Cambridge University Press, Cambridge, United Kingdom.

[12] C. J. COLBOURN, 2002. Projective planes and congestion-free networks, *Discrete Applied Mathematics 122*, 117–126.

[13] C. J. COLBOURN and J. H. DINITZ, (eds.), 1996. *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL.

[14] C. J. COLBOURN, J. H. DINITZ, and D. R. STINSON, 2001. Quorum systems constructed from combinatorial designs, *Information and Computation 169*, 160–173.

[15] C. J. COLBOURN and R. MATHON, 1996. Steiner systems, in *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 66–75.

[16] C. J. COLBOURN and P. C. V. OORSCHOT, 1989. Application of combinatorial designs in computer science, *ACM Computing Surveys 21*, 223–250.

[17] W. J. COOK, W. H. CUNNINGHAM, W. R. PULLEYBLANK, and A. SCHRIJVER, 1998. *Combinatorial Optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, New York, NY.

[18] G. CORNUÉJOLS and W. R. PULLEYBLANK, 1980. A matching problem with side conditions, *Discrete Mathematics 29*, 135–159.

[19] W. H. CUNNINGHAM and Y. WANG, 2000. Restricted 2-factor polytopes, *Mathematical Programming, Series A 87*, 87–111.

[20] R. DIESTEL, 2000. *Graph Theory*, Graduate Texts in Mathematics, Springer-Verlag, New York, NY, 2nd ed.

[21] J. H. DINITZ, *Handbook of Combinatorial Designs: New Results*. `http://www.emba.uvm.edu/~dinitz/newresults.html`. Accessed on March 15, 2004.

[22] J. H. DINITZ and D. R. STINSON, 1992. A brief introduction to design theory, in *Contemporary Design Theory: A Collection of Surveys*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, New York, NY, 1–12.

[23] A. FRANK, 2003. Restricted t-matchings in bipartite graphs, *Discrete Applied Mathematics 131*, 337–346.

[24] S. FURINO, 1996. The first time, in *Computational and Constructive Design Theory*, W.D. Wallis (ed.), Mathematics and Its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 13–28.

[25] M. R. GAREY and D. S. JOHNSON, 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, NY.

[26] K. GOPALAKRISHNAN and D. R. STINSON, 1996. Applications of designs to cryptography, in *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 549–557.

[27] M. GRÓTSCHEL and L. LOVÁSZ, 1995. Combinatorial optimization, in *Handbook of Combinatorics*, *Volume II*, R. L. Graham, M. Grótschel, and L. Lovász, (eds.), The MIT Press, Cambridge, MA, 1541–1598.

[28] H. HANANI, 1961. The existence and construction of balanced incomplete block designs, *Annals of Mathematical Statistics 32*, 361–386.

[29] D. HARTVIGSEN, 1999. The square-free 2-factor problem in bipartite graphs, in *Proceedings of the 7th International IPCO Conference*, G. Cornuéjols, R. Burkard, and G. Woeginger, (eds.), Volume 1610 of Lecture Notes in Computer Science, Springer-Verlag, Graz, Austria, 234–241.

[30] J. HERMAN, R. KUČERA, and J. ŠIMŠA, 2003. *Counting and Configurations: Problems in Combinatorics, Arithmetic, and Geometry*, CMS Books in Mathematics, Springer-Verlag, New York, NY.

[31] S. HOUGHTEN, L. THIEL, J. JANSSEN, and C. LAM, 2001. There is no (46,6,1) block design, *Journal of Combinatorial Designs 9*, 60–71.

[32] IBM CORPORATION, 1995. *OSL Optimization Solutions Library Guide and Reference*, New York, NY.

[33] A. ITAI, M. RODEH, and S. L. TANIMOTO, 1978. Some matching problems for bipartite graphs, *Journal of the ACM 25*, 517–525.

[34] P. KASKI and P. ÖSTERGARD, 2001. There exists no (15,5,4) RBIBD, *Journal of Combinatorial Designs 9*, 357–362.

[35] Z. KIRÁLY, 2001. C_4-free 2-factors in bipartite graphs, Technical Report TR-2001-13, Egerváry Research Group on Combinatorial Optimization (EGRES), Eötvös University, Budapest, Hungary.

[36] B. KORTE and J. VYGEN, 2002. *Combinatorial Optimization: Theory and Algorithms*, Algorithms and Combinatorics 21, Springer-Verlag, Berlin, Germany, 2nd ed.

[37] D. L. KREHER, 1996. t-designs, $t \geq 3$, in *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 47–66.

[38] C. MANNINO and A. SASSANO, 1995. Solving hard set covering problems, *Operations Research Letters 18*, 1–5.

[39] F. MARGOT, 2003. Small covering designs by branch-and-cut, *Mathematical Programming, Series B 94*, 207–220.

[40] R. MATHON and A. ROSA, 1996. $2-(v, k, \lambda)$ designs of small order, in *Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 3–41.

[41] W. H. MILLS and R. C. MULLIN, 1992. Coverings and packings, in *Contemporary Design Theory: A Collection of Surveys*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, New York, NY, 371–399.

[42] D. C. MONTGOMERY, 1984. *Design and Analysis of Experiments*, Wiley, New York, NY, 2nd ed.

[43] L. MORALES, 2000. Constructing difference families through an optimization approach: Six new BIBDs, *Journal of Combinatorial Designs 8*, 261–273.

[44] L. MORALES and C. VELARDE, 2001. A complete classification of (12,4,3)-RBIBDs, *Journal of Combinatorial Designs 9*, 385–400.

[45] L. MOURA, 1996. Polyhedral methods in design theory, in *Computational and Constructive Design Theory*, W.D. Wallis (ed.), Mathematics and Its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 227–254.

[46] L. MOURA, 1999. A polyhedral algorithm for packings and designs, in *ESA'99*, J. Něsetřil, (ed.), Volume 1643 of Lecture Notes in Computer Science, Springer-Verlag Berlin, 462–475.

[47] L. MOURA, 1999. *Polyhedral Aspects of Combinatorial Designs*, Ph.D. Thesis, University of Toronto, Toronto, Canada.

[48] L. MOURA, 2003. Rank inequalities and separation algorithms for packing designs and sparse triple systems, *Theoretical Computer Science 297*, 367–384.

[49] G. L. NEMHAUSER and L. A. WOLSEY, 1988. *Integer and Combinatorial Optimization*, Wiley, New York, NY.

[50] F. PIPER and P. WILD, 1992. Incidence structures applied to cryptography, *Discrete Mathematics 106*, 383–389.

[51] A. SCHRIJVER, 1986. *Theory of Linear and Integer Programming*, Wiley, Chichester, United Kingdom.

[52] D. R. STINSON, 1988. A construction for authentication/secrecy codes from certain combinatorial designs, in *Advances in Cryptology*, Volume 293 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany, 355–366.

[53] D. R. STINSON, 1996. Coverings, in *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 260–265.

[54] D. R. STINSON, 1996. Packings, in *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 409–413.

[55] V. D. TONCHEV, 1988. *Combinatorial Configurations: Designs, Codes and Graphs*, Longman Scientific and Technical, Wiley, New York, NY.

[56] V. D. TONCHEV, 1996. Codes, in *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 517–543.

[57] G. VAN REES, 1996. (22,33,12,8,4)-BIBD, an update, in *Computational and Constructive Design Theory*, W.D. Wallis (ed.), Mathematics and Its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 337–357.

[58] T. VAN TRUNG, 1996. Symmetric designs, in *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 75–87.

[59] W. D. WALLIS, 1988. *Combinatorial Designs*, Monographs and Textbooks in Pure and Applied Mathematics, Marcel Dekker, New York, NY.

[60] W. E. WILHELM, 2001. A technical review of column generation in integer programming, *Optimization and Engineering 2*, 159–200.

[61] R. M. WILSON, 1975. An existence theory for pairwise balanced designs, iii. proof of the existence conjectures, *Journal of Combinatorial Theory, Series A 18*, 71–79.

[62] R. M. WILSON, 1983. Inequalities for t designs, *Journal of Combinatorial Theory, Series A 34*, 313–324.

[63] L. A. WOLSEY, 1998. *Integer Programming*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, New York, NY.

[64] V. ZINOVIEV, 1996. On the equivalence of certain constant weight codes and combinatorial designs, *Journal of Statistical Planning and Inference 56*, 289–294.

VITA

Ivette Arámbula Mercado was born in Guadalajara, Mexico in 1970 to Martín Arámbula and Lilia R. Mercado. She graduated from H.S. in 1987 receiving the highest honor. She holds a B.S. in chemical engineering and management from Tecnológico de Monterrey (ITESM) obtained with honors in 1992. She holds a M.S. in industrial engineering and operations research obtained also from ITESM in 1995. She was a recipient of a CONACYT scholarship to pursue her master's degree, and during those studies she worked as research assistant at the ITESM Center for Strategic Studies, and as a lecturer for the ITESM Mathematics Department. She was a member of the junior faculty at ITESM from 1995 to 1999, lecturing Mathematics and Industrial Engineering courses. Also, from 1996 to 1998, she worked as an independent operations research consultant for two airlines in Mexico. In August 1999, she started the Ph.D. program at the Industrial Engineering Department of Texas A&M University in College Station, and received the Ph.D. in May 2004. During her doctoral studies, she held positions as research assistant, teaching assistant and lecturer. She worked on research in collaboration, separately, with professors Dr. Wilbert E. Wilhelm, Dr. Illya Hicks, and Dr. Jianer Chen. Her research interests are in integer and combinatorial optimization, large-scale optimization, and operations research. Ms. Arámbula Mercado can be reached at her permanent address:

Saltillo 1807
Mitras Centro
Monterrey, N.L. 64460
MEXICO

This document was typeset in LaTeX by the author.