Restricted b-Factors in Bipartite Graphs and t-Designs†

Ivette Arámbula,[1] Illya V. Hicks[2]
[1]*PROS Revenue Management, Houston, TX,*
E-mail: iarambula@prosrm.com

[2]*Industrial Engineering Department, Texas A&M University, College Station, TX. E-mail: ivhicks@tamu.edu*

Received April 9, 2004; revised September 12, 2005

Published online 15 February 2006 in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/jcd.20094

Abstract: We present a new equivalence result between restricted *b*-factors in bipartite graphs and combinatorial *t*-designs. This result is useful in the construction of *t*-designs by polyhedral methods. We propose a novel linear integer programming formulation, which we call GDP, for the problem of finding *t*-designs that has a noteworthy advantage compared to the traditional set-covering formulation. We analyze some polyhedral properties of GPD, implement a branch-and-cut algorithm using it and solve several instances of small designs to compare with another point-block formulation found in the literature. J Combin Designs 14: 169–182, 2006

Keywords: *combinatorial t-designs; integer programming; bipartite graphs; polyhedral combinatorics; branch-and-cut*

1. INTRODUCTION

A b-factor is a special case of the general matching problem in a graph. It involves choosing a subset of edges of the graph, subject to degree equality constraints on the vertices, and allowing each edge to be chosen no more than once. The term *restricted* refers to the requirement that the b-factor be free from certain induced 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 polyhedron associated with the case $k = 3$, the *triangle-free 2-factor* problem, was studied by Cornuéjols and Pulleyblank [8] because of its relation to the travelling salesman problem. Cunningham and Wang [9]

---

†Ivette Arámbula is a scientist and Illya V. Hicks is an assistant professor.
Contract grant sponsor: NSF; Contract grant number: DMI-0217265.

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* problem was studied by Hartvigsen [12] for bipartite graphs.

A t-design, denoted t-(v, k, λ) design, is an incidence structure with the properties that there are v points; every block is incident with k points; and any t points are incident with λ blocks. From the definition, it is clear that a t-design satisfies some strict properties of size, balance, and replication. This highly structured system provides optimal solutions to a variety of problems in computer science [6], like error-correcting codes [27,28], cryptography [23,25], and network interconnection [3,4], to mention a few. Special cases of t-designs include block designs, Steiner systems, triple and quadruple systems, projective geometries, affine geometries, and Hadamard designs. The main problem in t-designs is the question of existence and the construction of those solutions, given admissible parameters (discussed later). For example, the construction of a 6-$(v,k,1)$ design (smallest t for unknown Steiner systems) remains one of the outstanding open problems in the study of t-designs [11].

On the subject of polyhedral methods for solving b-factor problems, there exist linear-programming descriptions as well as results on polynomial time solvability [7]. 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 [8]) proved that deciding if a graph has a 2-factor with no cycle of length $k=5$ or less is NP-complete.

The construction of designs by polyhedral methods has been studied by Moura [19–21], Margot [16,17], indirectly by Mannino and Sassano [15], and recently by Arámbula [1]. The most natural integer programming formulation used for the problem of finding a t-design is a set-packing or set-covering type of formulation. This formulation for a t-(v, k, λ) design, has as many variables as the number of subsets of size k out of v points, that is $\binom{v}{k}$ variables. The number of constraints is $\binom{v}{t}$. This way to represent a t-design is known as *tk-incidence* representation. Another way to represent a t-design is the *point-block incidence* representation that will be defined later. An integer programming formulation for the special case of 2-designs that uses the point-block representation is mentioned by Moura [20]. This formulation is called QDP and has quadratic constraints that could be generalized to constraints of order t for handling t-designs. According to Moura [20], Wengrzik proposed and studied a cutting-plane algorithm on a linearized version of the quadratic constraints of QDP called LDP for the case of 2-designs.

We contribute by showing that the problem of finding a combinatorial t-design is equivalent to the problem of finding a biclique-free b-factor in a complete bipartite graph. This equivalence results leads to a novel integer programming formulation for the problem of finding a t-design, which we call GDP. GDP uses the point-block incidence matrix representation of a t-design. The point-block matrix was also the one used by Wengrzik (see [20]) for LDP; the two formulations differ on a number of constraints.

In the literature, most of the 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-$(v, k, 1)$. Only a few polyhedral studies have been pursued, and those for specific cases of the problem. GDP has a remarkable advantage compared to the set-covering or set-packing formulation that it can handle t-designs with any number of repeated blocks without the need of changing the binary variables into general integer variables. GDP has also the advantage of not requiring quadratic or non-linear constraints as another formulations reported in the literature. Other advantage of GDP is that it is a result of a problem equivalence. The number of variables in GDP is the smallest among all models, but the number of constraints is very large. However, it is suitable for a branch-and-cut approach.

This paper is organized as follows. In Section 2, we present some definitions from graph theory, the formal definition of a t-design, and some theorems from design theory. In Section 3, we state the equivalence result and its proof. In Section 4, we present the integer programming formulation GDP, give some remarks on the polyhedron associated with the relaxation of GDP including its dimension. Finally, in Section 5 we describe the implementation of a branch-and-cut algorithm on GDP, including computational results comparing generalized LPD and GDP for some test instances.

2. PRELIMINARIES

For the purpose of this manuscript we will assume that all graphs are simple and undirected. The reader is referred to an excellent book by Diestel [10] for unfamiliar terms not defined in this section.

Definition 2.1. *For a graph $G = (V, E)$ and $A \subseteq V$, a cut $\delta(A)$ of G is a set defined as $\delta(A) = \{e \in E : e \text{ has an end in } A \text{ and an end in } V \backslash A\}$. For a single vertex $v \in V$, the cardinality of its cut $|\delta(v)|$, is called degree of the vertex v.*

Definition 2.2. *A graph $G = (V, E)$ is called complete if all the vertices of G are pairwise adjacent.*

Definition 2.3. *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.*

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 non-negative integer number of times [22]. 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 2.4. *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 2.5. *A biclique is a complete bipartite graph, denoted $K_{p,q}$, where p is the number of vertices in one class and q the number of vertices in the other class. Trivial biclique graphs of the form $K_{p,1}$ or $K_{1,q}$ are called* stars.

The formal definition of a t-design is as follows:

Definition 2.6 [14]. *A t-(v, b, r, k, λ) design, or t-design, is a pair $(X, \mathcal{B})$ where $\mathcal{B}$ is a collection of b sets (called blocks) each of size k from the ground set X of v elements* (*called points*), *such that every point occurs r times, and every subset of t points of X occurs exactly* λ *times in* $\mathcal{B}$ $(t, v, b, r, k, \lambda \in \mathbb{Z}_+,\ t < k < v,\ \lambda > 0)$.

Usually, an instance of a *t*-design is defined giving only the parameters t-(v, k, λ) since b and r can be computed from the other parameters using the following theorem:

Theorem 2.7 [2,5,26]. *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*

$$\lambda_s = |\{B \in \mathcal{B}\} : S \subseteq B\}| = \frac{\lambda\binom{v-s}{t-s}}{\binom{k-s}{t-s}}, \qquad 0 \leq s \leq t. \tag{1}$$

In particular, $b = \lambda_0,\ r = \lambda_1,\ \lambda = \lambda_t$. Since λ_s in equation (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. Other remark about the admissible parameters is that without loss of generality one can consider only t-(v, k, λ) designs with $t < k < v/2$ because the design on the *supplement* of $\mathcal{B}$, $\mathcal{B}' = \{X \setminus B : B \in \mathcal{B}\}$ is a t-$(v, v-k, \lambda')$ design where $\lambda' = \lambda\binom{v-t}{k}/\binom{v-t}{k-t}$ [14]. For the rest of this paper these will be the assumptions on the admissible parameters.

From Theorem 2.7, we have the following,

Corollary 2.8 [26]. *A t-design is also a s-design for* $0 \leq s \leq t$.

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*. In this work, we focus on the later, which is defined as follows:

Definition 2.9. *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.*

The representation of a *t*-design that will be used in the equivalence and later in the new formulation is the point-block incidence matrix.

3. EQUIVALENCE RESULT

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 $b : V(G) \to \{r, k\}$ defined as,

$$b(v) = \begin{cases} r, & \text{if } v \in L \\ k, & \text{if } v \in R \end{cases} \tag{2}$$

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. Our equivalence result is the following:

Theorem 3.1. *Given admissible parameters t, v, k,* λ*, b, r, a t-(v,k,*λ*) design with b blocks and point replication r is equivalent to 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 will show $\mathcal{T} = \mathcal{F}$.

First, we need to 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 an edge $e \in E(G)$ if $(x, B) \in I$ for any $x \in X$, $B \in \mathcal{B}$ (see Figures 1 and 2 for a particular 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, we need to show $\mathcal{F} \subseteq \mathcal{T}$. For any $F \in \mathcal{F}$, the restriction to be a $\{r, k\}$-regular graph 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 $F = G(L \cup R, E)$. 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. Thus, to complete the proof we need to show that any t-set of points is incident to exactly λ blocks. This is done by induction on t.

When $t = 1$ this true since $\lambda_1 = r$. Thus, we may assume the claim is true when $t \leq \alpha$ where $\alpha \geq 1$. Let $t = \alpha + 1$. Pick a vertex $u \in L$ and consider the subgraph H induced by $(L \setminus \{u\}) \cup N(u)$ (i.e., $H = F[(L \setminus \{u\}) \cup N(u)]$).

We claim that H is a $\{\lambda_2, k-1\}$-factor. To see that, suppose H is not a $\{\lambda_2, k-1\}$-factor. By the construction of H, every vertex of $N(u)$ has degree $k - 1$ in H. Thus, there must exist a vertex w with degree greater than λ_2 since $\lambda_2(v-1) = r(k-1)$. Thus, F contains a K_{2,λ_2+1}. Let Q denote $N(u) \cap N(w)$ in F. Also, for a vertex $v \in Q$, v has $k - 2$ edges to choose $t - 2$ vertices other than u and w to form a t-set that includes u and w. However, by the pigeon-hole principle there exists a $(t-2)$-set from L (excluding u and w) that is adjacent to at least c members of Q where $c = \lceil (|Q|\binom{k-2}{t-2}/\binom{v-2}{t-2} \rceil$. Since $|Q| \geq \lambda_2 + 1$, $c \geq \lambda + 1$ which is a contradiction.

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

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

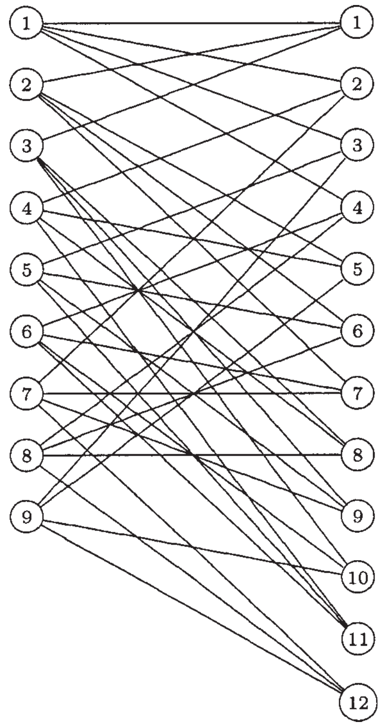

FIGURE 2. A biclique $K_{2,2}$-free $\{4,3\}$-factor in a bipartite graph $K_{9,12}$.

Then H is a $\{\lambda_2, k-1\}$-factor that is $K_{t-1,\lambda+1}$-free. By induction, H is equivalent to a $(t-1)$-$(v-1, k-1, \lambda)$ design. Thus, every $(t-1)$-subset of $V(H)$ is incident to exactly λ blocks in H, which implies that every t-subset of L that includes u is incident to exactly λ blocks in F. Since u was chosen arbitrarily, any t-set of points is incident to exactly λ blocks and the proof is complete. □

Note that if we translate the graph theoretical statements into design statements, this result parallels the fact that a t-(v, k, λ) packing with $b = \lambda\binom{v}{t}/\binom{k}{t}$ blocks is equivalent to a t-(v, k, λ) design.

This equivalence result will be used to propose a new integer programming formulation for t-design problems, that will be useful in constructing solutions for these problems by polyhedral methods. The equivalence result also has the following corollary.

Corollary 3.2. *Given admissible parameters t, v, k, λ, b, r, if the bipartite graph* $G = (L \cup E, E)$ *where* $|L| = v$ *and* $|R| = b$ *is a* $K_{t,\lambda+1}$*-free* r, k*-factor then G is also* K_{s,λ_s+1}*-free for* $0 \leq s \leq t$*.*

4. GDP: NOVEL INTEGER PROGRAMMING FORMULATION

For our integer linear programming formulation for t-designs, we use the following notation:

Notation 4.1. *For a graph $G = (V, E)$ and $M \subseteq E(G)$, $x(M) = \sum_{e \in M} x_e$.*

Let x be an incidence vector of a t-design using the point-block representation. Then $x \in \{0,1\}^{vb}$, 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 element 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:

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

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, that is, that every t-set of points appears together in exactly λ blocks of the design. Note that in the trivial case of $t = 1$ the biclique constraints become redundant with the point-star constraints. This reduces to a b-factor problem in bipartite graph for which efficient solution methods exist as mentioned in Section 1.

For the rest of this section, we will analyze the polyhedral aspects of GDP considering the linear programming relaxation of it. That is, the problem GDP in which $x_e \in \{0,1\}$ is substituted by $0 \leq x_e \leq 1$, that will be called GDP^R.

We will analyze the polyhedron associated with GDP^R as the intersection of an integral polyhedron with a non-integral polyhedron and show that it is not full-dimensional.

The polyhedron P_{GDP^R} is a bounded polyhedron (polytope) that can be written of the form:

$$P_{GDP^R} = \{x \geq 0 : Ax \leq d\} \tag{3}$$

For simplicity of notation, let $u = t(\lambda + 1) - 1$. The constraint matrix A and the right-hand side vector d can be written partitioned as follows:

$$P_{GDP^R} = \{x \geq 0 : \begin{bmatrix} A' \\ -A' \\ A'' \\ -A'' \\ A''' \\ I_{vb} \end{bmatrix} x \leq \begin{pmatrix} r \\ -r \\ k \\ -k \\ u \\ 1 \end{pmatrix} \} \tag{4}$$

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

$$A' = \begin{bmatrix} 1^T & 0 & \dots & 0 \\ 0 & 1^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1^T \end{bmatrix} \tag{5}$$

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

$$A'' = [\, I_b \quad I_b \quad I_b \quad \dots I_b \,] \tag{6}$$

and 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 subgraphs induced by all $\{t, \lambda+1\}$-bicliques.

Another way to view P_{GDP^R} is as the intersection of two polyhedra, one with a special property. Let $P_{biclique}$ be the polyhedron associated with the biclique constraints and P_{star} the polyhedron associated with the star constraints. That is,

$$P_{star} = \{x \geq 0 : A'x \leq r, -A'x \leq -r, A''x \leq k, -A''x \leq -k\}$$

and

$$P_{biclique} = \{x \geq 0 : A'''x \leq u\}$$

Definition 4.1. *A matrix A is totally unimodular if each subdeterminant of A is* 0, +1 *or* −1.

It is a well known result from Hoffman and Kruskal (see [7,13,24]) that a polyhedron of the form Equation (3) is integral for each integral right-hand side vector d if only if A is totally unimodular. P_{star} is an integral polyhedron since $[A', -A', \ A'', -A'']^T$ is a totally unimodular matrix (see matrix representation Equations (5) and (6)), and the right-hand side vector is an integral vector by definition of parameters r and k. Moreover, the polyhedron $P_{star(0-1)}$ including the upper bound on the variables

$$P_{star(0-1)} = \{x \in P_{star} : x \leq 1\}$$

is also integral due to the fact that also the matrix $[A' - A' \ A'' - A'' \ I]^T$ is totally unimodular.

Then another representation for the linear programming relaxation of GDP is

$$P_{GDP^R} = P_{star(0-1)} \cap P_{biclique} \tag{7}$$

which is the intersection of an integral polyhedron with a non-integral polyhedron.

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 [7]). So, to show that dim(P) $= n$, it suffices to find $n+1$ affinely-independent points in P.

Proposition 4.2. *The polytope $P_{biclique}$ is full dimensional.*

Proof. Let $x_i = (0, \dots, 1, \dots, 0)$ be an n-dimensional vector with a one in the ith position and zero elsewhere. Then, the $n+1$ points $x_1, \dots, x_n, 0$ satisfy the biclique inequalities and are affinely-independent. □

Another way to obtain the dimension of a polyhedron in n variables is by computing n minus the rank of the matrix of implicit equalities $A^=$. A polyhedron is thus full-dimensional if and only if there are no implicit equalities. In our case, the star constraints are clearly implicit equalities in P_{GDP^R}. To show that the biclique constraints are not implicit inequalities for non-trivial cases (i.e., $t > 1$) given admissible parameters take $x = (k/v, ..., k/v)$. The left-hand side of any biclique inequality for this x is $LHS = (k/v)t(\lambda + 1)$. Given the assumptions on admissible parameters that $v > 2k$ and $\lambda \geq 1$, we get

$$LHS < \frac{t(\lambda+1)}{2} = t(\lambda+1) - \frac{t(\lambda+1)}{2} < t(\lambda+1) - 1$$

so the biclique inequalities are strict. Then the dimension of P_{GDP^R} is equal to vb minus the rank of the matrix $A^=$, where v is the number of points in the design, b is the number of blocks in the design and $A^=$ is the matrix corresponding to the star inequalities. We conclude this section with the following result:

Proposition 4.3. *The dimension of the polytope associated with the linear programming relaxation of GDP is equal to:*

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

Proof. 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 Equations (5) and (6) that the rows within these two classes are linearly independent. However, the sum of the point-star rows can be obtained by the sum of the block-star rows. Therefore, $A^=$ is not full row rank, but number of rows minus one. □

5. BRANCH-AND-CUT IMPLEMENTATION AND COMPUTATIONAL RESULTS

Since our formulation is similar to LDP of Wengrzik (given below for general t) in the sense that both use the point-block incidence representation, our computational results consist of comparing GDP to LDP for a class of t-designs.

$$\begin{array}{lll} & \text{maximize}\, x(E) & \\ \text{s.t.} & x(\delta(v)) = k, & v \in Y \\ & \sum_{l=1}^{b} y_{Tl} = \lambda, & \forall t\text{-sets}\, T \\ & y_{Tl} - x_{il} \leq 0, & \forall i \in T, 1 \leq l \leq b, \forall t\text{-sets}\, T \\ & \sum_{i \in T} x_{il} - y_{Tl} \leq t - 1, & \forall t\text{-sets}\, T, 1 \leq l \leq b \\ & x_e, e \in E, y_{Tl} \in \{0,1\}, & \forall t\text{-sets}\, T, 1 \leq l \leq b \end{array} \qquad \text{(LDPgen)}$$

We implemented a simple branch-and-cut algorithm for the two formulations programmed in C++ and utilizing the CPLEX 8.1 callable library where CPLEX was only used to solve LP formulations. All computations were conducted on a 500 MHz Origin 3800. We utilized a warm start for both formulations as suggested by a referee for fixing variables which consists of fixing the edges of the first point and the edges of all blocks passing through a $(t-1)$-set including the first point plus one additional block when $\lambda = 1$; and fixing the edges of one block plus the edges of the first two points when $\lambda > 1$. In addition, since the edges of the first point are known for any λ, we added the equations that forces any point other than the first point to be adjacent to only λ_2 of the blocks that are fixed to be adjacent to the first point (similar to the construction of H in Theorem 3.1).

The details of the branch-and-cut algorithm are as follows:

- the GDP initial formulation for the root branch node consists of the point-star constraints, the block-star constraints, and the warm start equations;
- the LDP initial formulation for the root branch node consists of the block-star constraints, the addition of the $\sum_{l=1}^{b} y_{Tl} = \lambda$ constraints, and the warm start equations;
- a cut leaves the formulation once its slack value is greater than or equal to one;
- branch nodes are kept in a priority queue such that the branch node *bn* with the largest number of edge decision variables equal to one in the known solution to its relaxation is visited first;
- at a branch node, we only add cuts that are violated by at least 0.25 until there are no such constraints to add;
- at a branch node, once a cut is found, the new formulation is re-optimized;
- the branching variable is the edge decision variable whose value is closest to 0.5

Additionally for GDP, we visit every t-set (enumeration) and sort the blocks according to their total sum of the values of their edges incident to the t-set. If $\lambda + 1$ blocks whose total weight violates the $t(\lambda + 1) - 1$ bound by 0.25 is found, then the corresponding biclique inequality is added to the formulation and we re-optimize the formulation. We use a similar enumeration algorithm to find violated inequalities for LDP.

The instances are small t-designs from Mathon and Rosa [18], and t-designs from Moura [19] where $t \leq 3$. We also compared the two formulations as v grows fixing t, k, and λ (Table I and II), as λ grows fixing k and v (Table III), and for different values of t (Table IV). All computations are offered in the following tables where "bnodes" denotes the number of branch nodes other than the root that were created during the search. Also, "cuts" specifies the total number of cuts generated during the search and computational times are given in seconds "sec" with a 7200 seconds time limit. In addition, the "found?"column details if a design was found for the particular parameters and method.

From Tables I, II, III, and IV, GDP outperformed LDP for most cases. For example, LDP could find the design for 2-(21,5,1) in about 1000 seconds compared to GDP, which took no more than 200 seconds.

Unfortunately, neither LDP nor GDP performed well for Steiner quadruple systems. Thus, more research is needed to find relevant inequalities for these formulations for those designs.

6. CONCLUSIONS AND FUTURE WORK

This work gives a new polyhedral approach to t-designs, which is a result of a problem equivalence we found with a graph problem. The resulting integer linear

Table I. Computational Results for 2-(v,3,1)

		LDP				GDP			
v	b	found?	cuts	bnodes	sec	found?	cuts	bnodes	sec
7	7	yes	79	0	0.0	yes	6	0	0.0
9	12	yes	417	8	0.7	yes	50	2	0.1
13	26	yes	3828	264	189	yes	8035	520	50
15	35	yes	7389	389	946	yes	33366	612	314
19	57	no	19670	387	>7200	no	284595	3696	>7200

programming formulation has a remarkable advantage compared to the well, known set-covering or set-packing formulation. GDP can handle t-designs with any number of repeated blocks without the need of changing the binary variables into general integer variables. We believe that this equivalence result may open the possibility of further research such as a proof for the complexity of the problem that, to the best of our knowledge, has not been reported in the literature.

TABLE II. Computational Results for 3-(v,4,1)

		LDP				GDP			
v	b	found?	cuts	bnodes	sec	found?	cuts	bnodes	sec
8	14	yes	3626	1213	139	yes	40	0	0.1
10	30	no	12650	1627	>7200	yes	1862	86	14
14	91	no	42637	20	>7200	no	159083	130	>7200

TABLE III. Computational Results for Varying λ

	LDP				GDP			
t-(v, k, λ)	found?	cuts	bnodes	sec	found?	cuts	bnodes	sec
2-(7,3,1)	yes	79	0	0.0	yes	6	0	0.0
2-(7,3,2)	yes	299	0	0.4	yes	52	4	0.1
2-(7,3,3)	yes	527	2	1.1	yes	56	2	0.2
2-(7,3,4)	yes	669	3	1.7	yes	112	16	0.5
2-(7,3,5)	yes	988	6	4.1	yes	242	62	0.9
2-(13,4,1)	yes	1431	45	16	yes	133	0	0.7
2-(13,4,2)	no	4630	1422	>7200	no	6319	700	>7200
2-(21,5,1)	yes	7300	104	1205	yes	8870	152	143
2-(21,5,2)	no	18990	0	>7200	no	126029	0	>7200
3-(8,4,1)	yes	3626	1213	139	yes	40	0	0.1
3-(8,4,2)	no	4437	3255	>7200	yes	687	130	3.5
3-(8,4,3)	no	9276	2217	>7200	no	3716	761	>7200

Table IV. Computational Results for *t*-Designs

	LDP				GDP			
t-(v, k, λ)	found?	cuts	bnodes	sec	found?	cuts	bnodes	sec
2-(6,3,2)	yes	164	2	0.1	yes	12	2	0.0
2-(8,4,3)	yes	430	0	0.8	yes	94	10	0.2
2-(10,4,2)	yes	1389	127	43	yes	958	165	3.5
2-(11,5,2)	yes	1168	36	21	yes	662	115	2.6
2-(13,4,1)	yes	1431	45	16	yes	133	0	0.7
2-(16,4,1)	yes	4207	1340	5845	no	15265	834	>7200
3-(5,4,2)	yes	27	0	0.0	yes	0	0	0.0
3-(6,4,3)	yes	309	0	0.4	yes	11	0	0.0
4-(11,5,1)	no	67743	10	>7200	yes	24353	48	690

Our formulation GDP is a nice alternative over the two other existing formulations since it has a small number of variables. On the other hand, the number of rows in our formulation is very large, but it is suitable for a branch-and-cut approach, as the computational results show. With our computational results we were able to replicate solutions for some small-parameter designs and show that that GDP consistently outperformed LDP for constructing these designs. GDP also establishes another successful approach to link design theory with polyhedral combinatorics which may lead to new results on polyhedral aspects of combinatorial designs that have not been explored.

Future research directions include finding more efficient ways to solve the separation problem and investigating the strength of the biclique inequalities. Another avenue for research is to generate new classes of valid inequalities in order to find a better approximation of the convex hull of combinatorial designs.

ACKNOWLEDGEMENTS

This research was partially supported by NSF grant DMI-0217265. The authors thank an anonymous referee for valuable comments to improve this manuscript.

REFERENCES

[1] I. Arámbula, A New Polyhedral Approach to Combinatorial Designs. Ph.D. thesis, Texas A&M University, College Station TX, USA, 2004.

[2] T. Beth, D. Jungnickel, and H. Lenz, Design Theory, Vol. I, II of Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, United Kingdom, 2nd edition, 1999.

[3] C. J. Colbourn, Multiple access communications using combinatorial designs. In: Theoretical Aspects of Computer Science, Vol. 2292 of Lecture Notes in Computer Science, G. Khosrovshahi (Editor), Berlin Heidelberg, Germany, 2002. Springer-Verlag, pp. 1–29.

[4] C. J. Colbourn, Projective planes and congestion-free networks. Discrete Appl Math, 122 (2002), 117–126.

[5] C. J. Colbourn and J. H. Dinitz (Editors), The CRC Handbook of Combinatorial Designs, CRC Press, Boca Raton, FL, 1996.

[6] C. J. Colbourn and P. C. V. Oorschot, Application of combinatorial designs in computer science. ACM Comput Surveys 21 (1989), 223–250.

[7] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, Combinatorial Optimization. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, New York, NY, 1998.

[8] G. Cornuéjols and W. R. Pulleyblank, A matching problem with side conditions. Discrete Math 29 (1980), 135–159.

[9] W. H. Cunningham and Y. Wang, Restricted 2-factor polytopes. Math Programming, Series A 87 (2000), 87–111.

[10] R. Diestel, Graph Theory. Graduate Texts in Mathematics. Springer-Verlag, New York, NY, 2nd edition, 2000.

[11] J. H. Dinitz and D. R. Stinson, A brief introduction to design theory. In: Contemporary Design Theory: A Collection of Surveys, J. H. Dinitz and D. R. Stinson (Editors), Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, New York, NY, 1992, pp. 1–12.

[12] D. Hartvigsen, The square-free 2-factor problem in bipartite graphs. In: Proceedings of the 7th International IPCO Conference, Vol. 1610 of Lecture Notes in Computer Science, G. Cornuéjols, R. Burkard, and G. Woeginger (Editors), Graz, Austria, 1999. Mathematical Programming Society, Springer-Verlag, Heidelberg, pp. 234–241.

[13] B. Korte and J. Vygen, Combinatorial Optimization: Theory and Algorithms. Algorithms and Combinatorics 21. Springer-Verlag, Berlin, Germany, 2nd edition, 2002.

[14] D. L. Kreher, t-designs, $t \geq 3$. In: The CRC Handbook of Combinatorial Designs, C. J. Colbourn and J. H. Dinitz (Editors), CRC Press, Boca Raton, FL, 1996, pp. 47–66.

[15] C. Mannino and A. Sassano, Solving hard set covering problems. Operations Research Letters 18 (1995), 1–5.

[16] F. Margot, Exploiting orbits in symmetric ILP. Math Program, Series B 98 (2003), 3–21.

[17] F. Margot, Small covering designs by branch-and-cut. Math Program, Series B 94 (2003), 207–220.

[18] R. Mathon and A. Rosa, $2-(v,k,\lambda)$ designs of small order. In: Handbook of Combinatorial Designs, C. J. Colbourn and J. H. Dinitz (Editors), CRC Press, Boca Raton, FL, 1996, pp. 3–41.

[19] L. Moura, A polyhedral algorithm for packings and designs. In: ESA'99, Vol. 1643 of Lecture Notes in Computer Science, J. Něsetřil (Editor), Springer-Verlag Berlin, 1999, pp. 462–475.

[20] L. Moura, Polyhedral Aspects of Combinatorial Designs. Ph.d. thesis, University of Toronto, Toronto, Canada, 1999.

[21] L. Moura, Rank inequalities and separation algorithms for packing designs and sparse triple systems. Theo Comput Sci 297 (2003), 367–384.

[22] G. L. Nemhauser and L. A. Wolsey, Integer and Combinatorial Optimization. Wiley, New York, NY, 1988.

[23] F. Piper and P. Wild, Incidence structures applied to cryptography. Discrete Math 106 (1992), 383–389.

[24] A. Schrijver, Theory of Linear and Integer Programming. Wiley, New York, NY, 1986.

[25] D. R. Stinson, A construction for authentication/secrecy codes from certain combinatorial designs. In: Advances in Cryptology, Vol. 293 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany, 1988, pp. 355–366.

[26] V. D. Tonchev, Combinatorial Configurations: Designs, Codes and Graphs. Longman Scientific and Technical. Wiley, New York, NY, 1988.

[27] V. D. Tonchev, Codes. In: The CRC Handbook of Combinatorial Designs, C. J. Colbourn and J. H. Dinitz (Editors), CRC Press, Boca Raton, FL, 1996, pp. 517–543.

[28] V. Zinoviev, On the equivalence of certain constant weight codes and combinatorial designs. J Stat Plann Inference 56 (1996), 289–294.