International Journal of Operations Research

International Journal of Operations Research Vol. 4, No. 3, 138−145 (2007)

Using Hyperstars to Create Facial-Defining Inequalities of General Binary Integer Programs

Kevin Hooker and Todd Easton*

School of Industrial and Manufacturing Systems Engineering, Kansas State University, Manhattan, Kansas

Received May 2006; *Revised September 2006*; *Accepted November 2006*

Abstract—Theoretical results relating to the facial structure of the general binary integer-programming polytope $conv(\{x \in \{0, 1\}^n : Ax \leq b\})$ where $A \in \mathbb{R}^{r\times n}$, and $b \in \mathbb{R}^r$ are presented. A conflict hypergraph is constructed and some induced hyperstars create valid inequalities of P^{BIP}. These inequalities are further shown to produce large dimensional faces. Some computational results show the benefit of using hyperstar inequalities for the project allocation problem.

Keywords—Hypergraphs, Hyperstars, Polyhedral theory, Integer programming, Project allocation problem

1. INTRODUCTION

General binary integer programs (GBIP) take the form max c^Tx subject to $Ax \leq b$ and $x \in \{0, 1\}^n$ where $A \in \mathbb{R}^{r\times n}$, and $b \in \mathbb{R}^r$. Let P be the feasible region of a GBIP, $P = \{x \in \{0, 1\}^n : Ax \leq b\}$. Then the general binary integer program polytope is the convex hull of P, which is denoted by $P^{BIP} = conv\{x \in \{0, 1\}^n : Ax \leq b\}$.

Valid inequalities or cutting planes have been extensively used to help solve integer programs. An inequality $\sum_{i=1}^{n} \alpha_i x_i \leq \beta$ is valid if every $x \in P$ satisfies the inequality. A valid inequality defines a face of dimension q if and only if there are exactly $q + 1$ affinely independent points in P that satisfy $\sum_{i=1}^{n} \alpha_i x_i = \beta$. An inequality is said to be facet-defining if and only if the dimension of the valid inequality's face is one less than the dimension of P^{BIP}. See Nemhauser and Wolsey (1988) for more background on polyhedral theory.

Numerous researchers have used conflict graphs (Atamtürk et al. (2000), Bixby and Lee (1994), Borndörfer (1997), Chvátal (1973), Pulleyblank and Edmonds (1975)) or conflict hypergraphs (Cornuejols and Sassano (1989), Easton et al. (2003), Euler, Junger, and Reinelt (1987); Nemhauser and Trotter (1974), Padberg (1973), Padberg (1980), Sassano (1989)) to create both valid inequalities and facet-defining inequalities. The majority of this research has focused on special integer programming polytopes, such as the multiple knapsack polytope $P^{MK} = conv\{x \in \{0, 1\}^n : Ax \leq b, A \in \mathbb{R}_+^{r\times n}\}$ or the set covering polytope $P^{SC} = conv\{x \in \{0, 1\}^n : Ax \leq b, A \in \{0, 1\}^{r\times n}\}$.

This paper describes a conflict hypergraph and defines an induced hyperstar, which creates valid inequalities of P^{BIP}. Section 2 describes the creation of this conflict hypergraph. Section 3 introduces the hyperstar inequality and provides some theoretical results. Computational results and extensions to the project allocation problem are contained in Section 4. The paper concludes with Section 5, which summarizes the results and provides some directions for future research.

2. CONFLICT HYPERGRAPHS

A graph $G = (V, E)$ consists of a set of vertices $V(G) = \{1, ..., n\}$ and a set of edges $E(G) = \{\{u, v\} : u, v \in V\}$. From a P^{BIP} instance, a typical conflict graph can be generated by defining a vertex for each variable. An edge $\{i, j\}$ is in the edge set if and only if there is no $x \in P$ such that $x_i = 1$ and $x_j = 1$. In other words, edges represent infeasible portions of the space. Atamürk et al. (2000) further extended the vertex set of a conflict graph to include vertices that represent variables equal to 0. This paper creates a conflict graph that consists of a vertex set that represents variables set to both 1 and 0.

A hypergraph $H = (V, E)$ consists of a set of vertices $V(H) = \{1, ..., n\}$ and a set of edges $E(H) = \{d_1, ..., d_l\}$, where $d_i \subseteq V(H)$ for all i. The definitions of a subhypergraph and an induced subhypergraph follow directly from their graph theoretic definitions. A uniform k-hypergraph is a hypergraph in which every edge has cardinality k. Throughout the remainder of this paper, every mention of a hypergraph is referring to a uniform k-hypergraph, and unless it causes ambiguity, $E(H)$ will be replaced with E and $V(H)$ with simply V. Berge (1973) is a good source of background information on both graphs and hypergraphs.

Given a GBIP, define the k-conflict hypergraph to be $H_k = (V, E)$ where $V = \{i: i = 1, ..., n\} \cup \{\bar{i} : i = 1, ..., n\}$. There is a direct correspondence between variable x_i and vertices i and $\bar{i}$. The edge set E is constructed using the

* Corresponding author's email: teaston@ksu.edu

following method: $e \in E$, if $|e| = k$ and every point in $\{0, 1\}^n$ with $x_i = 0$ for all $\bar{i} \in e$ and $x_i = 1$ for all $i \in e$ is not in P. One added restriction is that i and $\bar{i}$ will never be in the same edge as this implies that $x_i = 0$ and $x_i = 1$, which is an inconsistency and not an infeasibility. Finally, an edge e is minimal if $e\backslash\{i\} \notin E(H_{k-1})$ for all $i \in e$ and $e\backslash\{\bar{i}\} \notin E(H_{k-1})$ for all $\bar{i} \in e$. The following example demonstrates the construction of this new type of conflict hypergraph.

Example 2.1: Consider the feasible region of an integer program defined as:

$$\begin{aligned} 4x_1 - 4x_2 + 2x_3 + 2x_4 - 2x_5 + x_6 - x_7 &\le 4 \\ 4x_1 - 4x_2 - x_3 + 2x_4 + x_5 - 2x_6 + 2x_7 &\le 4 \\ 2x_1 - 2x_2 - x_5 - x_6 + x_7 &\le 1 \\ 2x_1 - 2x_2 + x_3 - x_6 + x_7 &\le 2 \\ x \in \{0, 1\}^7. \end{aligned}$$

The vertex set of H_4 is $V = \{1, 2, ..., 7, \bar{1}, \bar{2}, ..., \bar{7}\}$. H_4 contains the edge $\{1, \bar{2}, 3, 4\}$ because any point with $x_1 = 1$, $x_2 = 0$, $x_3 = 1$ and $x_4 = 1$ will not satisfy the first constraint. Some of the other edges in H_4 include: $\{1, \bar{2}, 3, \bar{5}\}$, $\{1, \bar{2}, 4, \bar{5}\}$, $\{1, \bar{2}, \bar{6}, 7\}$ and $\{1, \bar{2}, 3, \bar{6}\}$.

Observe that any single constraint $\sum_{j=1}^{n} a_{i,j} x_j \le b_i$ in a GBIP can be transformed into a knapsack constraint, which takes the form $\sum_{j=1}^{n} a_{i,j} x_j \le b_i$ where each $a_{i,j} \ge 0$ for some fixed $i \in \{1, ..., m\}$. The tranformation merely replaces x_j by $1 - x'_j$ whenever the $a_{i,j} < 0$. Furthermore, a set $C \subseteq \{1, 2, ..., n\}$ is a cover of the knapsack constraint if and only if $\sum_{j \in C} a_{i,j} > \beta$. Observe that covers of these knapsack constraints will be edges in the conflict hypergraph.

It is important to note that there may exist edges in H_k that do not correspond to a cover from any single inequality. For instance, the H_2 generated from $\{x \in \{0, 1\}^n : x_1 + x_2 - x_3 \le 1, x_1 + x_2 + x_3 - x_4 \le 1\}$ has the edge $\{1, 2\}$. Clearly, examining any single constraint will not generate this edge and so not all edges in GBIP correspond to covers of an individual constraint. Furthermore, determining the existance of a noncover edge in a GBIPs is an *NP*-hard problem.

3. HYPERSTARS

A graph with m vertices is called a star or fan, denoted by S_m, if there exists a relabeling of the vertex set such that $E = \{(1, i) : i = 2, ..., m\}$. Vertex 1 is called the hub or center and the other vertices are called blades or spokes. There are two natural ways to extend the definition of a fan or star to a hyperfan or hyperstar. An example of a hyperfan with 2 hub nodes and 4 blade nodes is $\{1, 2, 3, 4\}$ and $\{1, 2, 5, 6\}$. Observe that each edge contains all of the hub vertices $\{1, 2\}$ and each blade vertex $\{3, 4, 5, 6\}$ is in exactly one edge. An example of a hyperstar with two hub nodes and 4 blade nodes is $\{1, 2, 3, 4\}$, $\{1, 2, 3, 5\}$, $\{1, 2, 3, 6\}$, $\{1, 2, 4, 5\}$, $\{1, 2, 4, 6\}$ and $\{1, 2, 5, 6\}$. Notice that every edge contains all of the hub vertices and the edge set contains all possible combinations of blade vertices. This paper derives valid inequalities from hyperstars.

Formally, a hypergraph with m vertices, $S_{m,l,k}$, is a hyperstar if there exists a set $L \subseteq V(S_{m,l,k})$ with $|L| = l \ge 1$ such that $E(S_{m,l,k}) = \{d \subseteq \{1, ..., m\}: |d| = k \text{ and } L \subseteq d\}$ with $k > l \ge 1$. That is $S_{m,l,k}$ contains every edge of size k that contains the vertices of the hub, L. Thus, $|E(S_{m,l,k})| = \dfrac{(m-l)!}{(m-k)!(k-l)!}$.

Induced hyperstars from the conflict hypergraph can lead to face-defining inequalities. First, the vertices of an induced hyperstar naturally partition into four mutually exclusive sets. The hub vertices, L, partition into the sets S_L and $\bar{S}_L$ where S_L contains the nonbarred hub vertices and $\bar{S}_L$ contains the barred hub vertices. Similarly, the blade vertices, $V(S_{m,l,k})\backslash L$, partition into S_B and $\bar{S}_B$. To simplify the following proofs, for any $x \in P$ let c_U be the number of active vertices in $U \subseteq S_{m,l,k}$. That is

i) $c_{S_B} = |\{i \in S_B : x_i = 1\}|$,
ii) $c_{\bar{S}_B} = |\{i \in \bar{S}_B : x_i = 0\}|$,
iii) $c_{S_L} = |\{i \in S_L : x_i = 1\}|$ and
iv) $c_{\bar{S}_L} = |\{i \in \bar{S}_L : x_i = 0\}|$.

Now hyperstars can be shown to induce a valid inequality.

Lemma 3.1. Given a general integer-programming problem with conflict hypergraph H_k with $k \ge 2$, if $S_{m,l,k}$ is a hyperstar in H_k with hub L, then

$$(m-k+1)\left[\sum_{i \in S_L} x_i - \sum_{i \in \bar{S}_L} x_i\right] + \sum_{i \in S_B} x_i - \sum_{i \in \bar{S}_B} x_i$$
$$\le (m-k+1) * (l - |\bar{S}_L|) + k - l - 1 - |\bar{S}_B|$$

is a valid inequality of P^{BIP}.

Proof. First rewrite the inequality as

$$(m-k+1)\left[\sum_{i \in S_L} x_i - \sum_{i \in \bar{S}_L} (1-x_i)\right] + \sum_{i \in S_B} x_i + \sum_{i \in \bar{S}_B} (1-x_i)$$
$$\le (m-k+1)l + k - l - 1.$$

Let $f_I(x)$ be the value of the left hand side of the above inequality at any point $x \in P$ and the proof divides into two cases.

Case 1. Let $c_{S_L} + c_{\bar{S}_L} = l$. This implies that

$$\begin{aligned} f_T(x) &= (m-k+1)(c_{S_L} + c_{\bar{S}_L}) + c_{S_B} + c_{\bar{S}_B} \\ &= (m-k+1)l + c_{S_B} + c_{\bar{S}_B} \\ &\le (m-k+1)l + k - l - 1. \end{aligned}$$

The last inequality follows because, by the definition of a hyperstar, any point $x \in P$ can have at most $k - l - 1$ blade vertices active when every hub vertex is active.

Case 2. Let $c_{S_L} + c_{\bar{S}_L} < l$. Then $c_{S_L} + c_{\bar{S}_L}$ is bounded by $l - 1$. Also note that there are only $m - l$ nodes outside the hub. Thus it follows that

$$\begin{aligned} f_T(x) &= (m-k+1)(c_{S_L} + c_{\bar{S}_L}) + c_{S_B} + c_{\bar{S}_B} \\ &\le (m-k+1)(l-1) + m - l \\ &= (m-k+1)l + k - l - 1. \end{aligned}$$

Observe that H_4 from Example 2.1 creates a $S_{7,2,4}$ since $\{1, \bar{2}, 3, 4\}$, $\{1, \bar{2}, 3, \bar{5}\}$, $\{1, \bar{2}, 3, \bar{6}\}$, $\{1, \bar{2}, 3, 7\}$, $\{1, \bar{2}, 4, \bar{5}\}$, $\{1, \bar{2}, 4, \bar{6}\}$, $\{1, \bar{2}, 4, 7\}$, $\{1, \bar{2}, \bar{5}, \bar{6}\}$, $\{1, \bar{2}, \bar{5}, 7\}$, and $\{1, \bar{2}, \bar{6}, 7\}$ are edges. Therefore, the corresponding hyperstar inequality is

$$4x_1 - 4x_2 + x_3 + x_4 - x_5 - x_6 + x_7 \le 3.$$

Furthermore, observe that the point $(1, 0, \frac{1}{3}, \frac{1}{6}, \frac{2}{3}, \frac{1}{3}, 0)$ is a feasible point in the linear relaxation, but is cut off by the hyperstar inequality (3.5 > 3).

In a multiple knapsack problem with $|L| = 1$, a hyperstar inequality is equivalent to Padberg's (1, k) configurations (1980), which are facet-defining under certain conditions. In general binary integer programs, hyperstars can also generate large dimensional face-defining inequalities as the following result shows.

Theorem 3.2. Given a general integer-programming problem with conflict hypergraph H_k with $k \ge 2$, if $S_{m,l,k}$ is an induced hyperstar containing m vertices with l vertices in the hub such that $n \ge m \ge k > l + 1 \ge 2$, and if for each vertex $v \in S_{m,l,k} \backslash L$ there exists an edge $d \in E(S_{m,l,k})$ such that $v \in d$ and d is a minimal edge, then

$$\begin{aligned} &(m-k+1)\left[\sum_{i \in S_L} x_i - \sum_{i \in \bar{S}_L} x_i\right] + \sum_{i \in S_B} x_i - \sum_{i \in \bar{S}_B} x_i \\ &\le (m-k+1) * (l - |\bar{S}_L|) + k - l - 1 - |\bar{S}_B| \end{aligned}$$

defines a face of dimension at least $m - l - 1$.

Proof. We have shown that the hyperstar inequality is valid, so it remains to show that the face $F_T = \{x \in P : (m - k + 1)[\sum_{i \in S_L} x_i - \sum_{i \in \bar{S}_L} x_i] + \sum_{i \in S_B} x_i - \sum_{i \in \bar{S}_B} x_i = (m - k + 1) * (l - |\bar{S}_L|) + k - l - 1 - |\bar{S}|\}$ contains $m - l$ affinely independent points. These points will be constructed algorithmically by analyzing the components generated by the minimal edges in $E(S_{m,l,k})$. Define a graph $G = (V, E)$ with $V(G) = V(S_{m,l,k}) \backslash L$. If u', $v' \in (d \backslash L)$ where $d \in E(S_{m,l,k})$ and d is a minimal edge, then $\{u', v'\}$ is an edge in G. Split G into the connected components $C_1, \ldots, C_q$ and begin by constructing the points for component C_1.

Let $\{d_1, \ldots, d_p\}$ be the set of minimal edges that were used to create the C_1 component. Set $R = d_1$ and $D = \{d_2, \ldots, d_p\}$. Since $d_1 = \{v'_1, \ldots, v'_k\}$ is minimal, for each $v'_j \in d_1 \backslash L$, there exists a point in P that is also in F_T with vertex v'_j inactive, all the vertices in $d_1 \backslash \{v'_j\}$ active and every vertex in $V(S_{m,l,k}) \backslash d_1$ inactive.

Since C_1 is connected, if R equals $V(C_1)$, begin constructing points from C_2. Otherwise there exists an edge $d_i \in D$ such that $(d_i \cap R) \ne L$. If $d_i \backslash R$ is only a single vertex, v'_j, then there must exist a point in P with v'_j, all vertices in L and exactly $k - l - 2$ other vertices of d_i active and the other vertices in $V(S_{m,l,k})$ inactive. Such a point must exist as d_i is a minimal edge. On the other hand, if $|d_i \backslash R| \ge 2$, then for each $v'_j \in d_i \backslash R$, a point in P exists with v'_j inactive, each vertex in $d_i \backslash \{v'_j\}$ active, and each vertex in $V(S_{m,l,k}) \backslash d_i$ inactive. An additional $|d_i \backslash R|$ points have been created. Set $R = R \cup d_i$, $D = D \backslash d_i$ and include these new point(s) to the previously generated points. Repeat this process until $R = V(C_1)$, and then repeat for each component C_i. After completing the process, $m - l$ points in F_T have been constructed. Denote these points as $R = \{r_1, \ldots, r_{m-l}\}$.

To show that these points are affinely independent it is necessary and sufficient to show that $\sum_{i=1}^{m-l} \lambda_i r_i = 0$ and $\sum_{i=1}^{m-l} \lambda_i = 0$ is uniquely solved by $\lambda_i = 0$ for all $i = 1, \ldots, m - l$. Let R' be the matrix of the system of equations that are being solved ($R'\lambda = 0$). If $\bar{i} \in \bar{S}_B$, then subtract the last row of R' from the i^{th} row of R' and then multiply this row by −1. These two operations change every 0 in this row to a 1 and every 1 to a 0. Delete the final row from R' and call this new matrix R''. R'' has the property that if i^{th} is active in r_j, then the i^{th} element of the j^{th} column is 1 and 0, if not.

R'' has a block diagonal structure relative to the rows corresponding to the blade vertices (i.e. the only nonzero elements in R'' are located between rows ($\sum_{i=1}^{j} |C_i|$) + 1 and $\sum_{i=1}^{j+1} |C_i|$ where $j = 0, \ldots, q - 1$. Thus it suffices to show that the $|C_1|$ points generated by component C_1 are linearly independent in R''. Restricting ourselves to the first k columns leads to a cyclical permutation of $k - 1$ consecutive 1's. Since this cyclical permutation is over k rows and columns and the greatest common divisor of k and $k - 1$ is 1, these k points are linearly independent

(Hooker (2004)). The next points maintain this linear independence by a similar argument. If the point involves the case of only increasing the size of R by 1, then this row has exactly one 1 and it appears in the next column and is trivially independent. If $|d_i \backslash R| = s \geq 2$, then the next added points contain a consecutive cyclical permutation of $s-1$ ones over an $s \times s$ matrix in the next s rows and s columns. Since these are the only non zeros in these rows, adding these points to the previous points maintains the linearly independence property and therefore, the hyperstar inequality supports a face with dimension at least $m - l - 1$.

The restriction that $k > l + 1$ is necessary, since if $k = l + 1$, then every blade vertex would be in its own component in G and the only point that would be generated by the proof has every hub vertex active and every blade vertex inactive. The $S_{7,2,4}$ from Example 2.1 satisfies Theorem 3.2 and thus it defines a face of dimension at least 3.

Theorem 3.2 also assumes that there is no knowledge about how many of the blade vertices can be active when one or more of the hub vertices is inactive. Thus in the current inequality, if even just one of the hub vertices is inactive in a point, then all of the non-hub vertices can be active without violating the hyperstar inequality. It is natural to question whether or not this inequality can be improved. First, for each $i' \in L$ define

$k_{i'} = min\{\{ k'' \in \{k, ..., m-1\}$: the induced subhypergraph of $H_{k''}$ on $V(S_{m,l,k}) \backslash \{ i' \}$ is a hyperstar with hub $L \backslash \{ i' \}\} \cup \{m\}\}$.

That is, $k_{i'}$ is the smallest sized edge such that $H_{k_{i'}}$ contains an induced subhypergraph of $V(S_{m,l,k} \backslash \{ i' \})$ that is a hyperstar with hub $L \backslash \{ i' \}$. If no such hyperstar exists, then $k_{i'} = m$. Essentially, $k_{i'} - l$ determines the maximum number of non-hub vertices that are active in any point in P for which i' is the only inactive hub vertex. Now for each $i' \in L$ define $p_i = \max(k_{i'}, \left\lceil \frac{m-l}{2} \right\rceil + k - 1)$. Using this definition of p_i, a stronger hyperstar inequality is produced.

Lemma 3.3. Given a general integer-programming problem with conflict hypergraph H_k with $k > l \geq 1$, if $S_{m,l,k}$ is a hyperstar in H_k with hub L, then

$$\sum_{i \in S_L} (p_i - k + 1) x_i - \sum_{i \in \bar{S}_L} (p_i - k + 1) x_i + \sum_{i \in S_B} x_i - \sum_{i \in \bar{S}_B} x_i \leq \sum_{i \in S_L} (p_i - k + 1) + k - l - 1 - \left| \bar{S}_B \right|$$

is a valid inequality for P^{BIP}.

Proof. Since $L = S_L \cup \bar{S}_L$ the inequality can be rewritten as

$$\sum_{i \in S_L} (p_i - k + 1) x_i + \sum_{i \in \bar{S}_L} (p_i - k + 1)(1 - x_i) + \sum_{i \in S_B} x_i + \sum_{i \in \bar{S}_B} (1 - x_i) \leq \sum_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) + k - l - 1 .$$

Again the result is that anytime a variable is active, it will increase the value of the left hand side. Next let $f_T(x)$ be the value of the left hand side of the above inequality at any point $x \in P$ and the proof divides into the following three cases:

Case 1. Let $c_{S_L} + c_{\bar{S}_L} = l$. Following the same logic as Lemma 3.1

$$\begin{aligned} f_T(x) &= \sum\nolimits_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) + c_{S_B} + c_{\bar{S}_B} \\ &\leq \sum\nolimits_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) + k - l - 1. \end{aligned}$$

Case 2. Let $c_{S_L} + c_{\bar{S}_L} = l - 1$. Without loss of generality, assume that vertex $r' \in L$ is inactive in x. Thus it follows that

$$\begin{aligned} f_T(x) &= \sum\nolimits_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) - (p_{r'} - k + 1) + c_{S_B} + c_{\bar{S}_B} \\ &\leq \sum\nolimits_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) - (p_{r'} - k + 1) + k_{r'} - l \\ &\leq \sum\nolimits_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) - (p_{r'} - k + 1) + p_{r'} - l \\ &= \sum\nolimits_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) + k - l - 1. \end{aligned}$$

The first inequality follows from the definition of k_j and the second from the fact that $p_j \geq k_j$.

Case 3. Let $c_{S_L} + c_{\bar{S}_L} \leq l - 2$. Clearly then $c_{S_L} + c_{\bar{S}_L}$ will be bounded by $l - 2$, and $c_{S_B} + c_{\bar{S}_B}$ is at most $m - l$. Assume that $r', s' \in L$ are inactive in x. Thus

$$\begin{aligned} f_T(x) &\leq \sum\nolimits_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) - (p_{r'} - k + 1) \\ &\quad - (p_{s'} - k + 1) + c_{S_B} + c_{\bar{S}_B} \\ &\leq \sum\nolimits_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) - 2\left\lceil \frac{m-l}{2} \right\rceil + m - l \\ &\leq \sum\nolimits_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) \\ &\leq \sum\nolimits_{i \in S_L \cup \bar{S}_L} (p_i - k + 1) + k - l - 1. \end{aligned}$$

These inequalities follow because $(p_i - k + 1) \geq \left\lceil \frac{m-l}{2} \right\rceil$ by definition and $k - l - 1 \geq 0$.

This stronger hyperstar inequality can now be shown to support a face of greater dimension than the original hyperstar inequality.

Theorem 3.4. Let $S_{m,l,k}$ be an induced hyperstar from the conflict hypergraph H_k containing m vertices with l vertices in the hub, $n \geq m \geq k > l+1 \geq 2$ and p_i as defined above. If for each vertex $v \in S_{m,l,k}$ there exists an edge $d \in E(S_{m,l,k})$ such that $v \in d$ and d is a minimal edge, then

$$\sum_{i\in S_L}(p_i-k+1)x_i-\sum_{i\in \bar{S}_L}(p_i-k+1)x_i+\sum_{i\in S_B}x_i-\sum_{i\in \bar{S}_B}x_i \leq \sum_{i\in S_L}(p_i-k+1)+k-l-1-\left|\bar{S}_B\right|$$

defines a face of dimension at least $m - l - 1 + |\{i : k_i \geq \left\lceil \frac{m-l}{2} \right\rceil + k - 1\}|$.

Proof. Define the face $F_T = \{x \in P : \sum_{i\in S_L}(p_i-k+1)x_i - \sum_{i\in \bar{S}_L}(p_i-k+1)x_i + \sum_{i\in S_B}x_i - \sum_{i\in \bar{S}_B}x_i = \sum_{i\in S_L}(p_i-k+1) + k - l - 1 - \left|\bar{S}_B\right|\}$. To show that it contains $m - l + |\{i : k_i \geq \left\lceil \frac{m-l}{2} \right\rceil + k - 1\}|$ affinely independent points, the first $m - l$ points will be exactly the points used in Theorem 3.2. Then, for each hub vertex $j' \in L$ such that $k_{j'} \geq \left\lceil \frac{m-l}{2} \right\rceil + k - 1$, note that $p_{j'} = k_{j'}$. Thus, there exists a point in $P \cap F_T$ with vertex j' inactive, each vertex in $L\backslash\{j'\}$ active, and exactly $p_{j'} - l$ blade vertices active, which gives $|\{i' : k_{i'} \geq \left\lceil \frac{m-l}{2} \right\rceil + k - 1\}|$ additional affinely independent points in the face.

A clear deduction from the above result is that if $p_{i'} = k_{i'}$ for all $i' \in L$, then the above hyperstar inequality defines a face of dimension at least $m - 1$.

By imposing a maximality condition on a hyperstar, we can show that all other variables will sequentially lift into a hyperstar inequality with a 0 coefficient. First the concept of a maximal hyperstar must be well-defined. A hyperstar is said to be maximal if there does not exist a hyperstar by moving a blade vertex to the hub vertex set or by adding any vertex not in the hyperstar to either the hub or blade vertex sets. With this definition the following result is attainable:

Theorem 3.5. Let $S_{m,l,k}$ be a maximal hyperstar with hub L and $n \geq m \geq k > l+1 \geq 2$ in the conflict hypergraph H_k, and let $x_j \in V(H_k)\backslash V(S_{m,l,k})$. Then the following two statements are true:

1 Either there is no point in P with $x_j = 0$ or there exists a point with $x_j = 0$ and

$$\sum_{i\in S_L}(p_i-k+1)x_i-\sum_{i\in \bar{S}_L}(p_i-k+1)x_i+\sum_{i\in S_B}x_i-\sum_{i\in \bar{S}_B}x_i = \sum_{i\in S_L}(p_i-k+1)+k-l-1-\left|\bar{S}_B\right|.$$

2 Either there is no point in P with $x_j = 1$ or there exists a point with $x_j = 1$ and

$$\sum_{i\in S_L}(p_i-k+1)x_i-\sum_{i\in \bar{S}_L}(p_i-k+1)x_i+\sum_{i\in S_B}x_i-\sum_{i\in \bar{S}_B}x_i = \sum_{i\in S_L}(p_i-k+1)+k-l-1-\left|\bar{S}_B\right|.$$

Hence, x_j is sequentially lifted with a 0 coefficient into the hyperstar inequality, i.e. the lifting does not change a maximal hyperstar inequality.

Proof. In either case, $x_j = 0$ or $x_j = 1$, if there does not exist a point in P, the proof is trivial. To show Case 1, suppose the maximal hyperstar meets the above conditions, that $x \in P$ with $x_j = 0$, and that

$$\sum_{i\in S_L}(p_i-k+1)x_i-\sum_{i\in \bar{S}_L}(p_i-k+1)x_i+\sum_{i\in S_B}x_i-\sum_{i\in \bar{S}_B}x_i < \sum_{i\in S_L}(p_i-k+1)+k-l-1-\left|\bar{S}_B\right|.$$

Then there does not exist a point with vertex $\bar{j}$ active, all the vertices in L active, and $k - l - 1$ vertices of $V(S_{m,l,k})\backslash L$ set active in P. Thus, vertex $\bar{j}$ can be added to the hyperstar as an additional blade vertex, and the assumption that $S_{m,l,k}$ is maximal is contradicted. The proof is similar for Case 2.

It may seem that Theorem 3.5 would imply that a maximal hyperstar inequality would define a facet of P^{BIP}. However, this not the case as the face of the hyperstar inequality may define the whole feasible space.

Observe that the property that no blade vertex can be moved into the hub was not utilized in the proof of this theorem. Since Theorem 3.5 only lifts variables in $V\backslash V(S_{m,l,k})$ this assumption is not needed. However, if a blade vertex could actually be a hub vertex, then its coefficient in the hyperstar inequality could be strengthened. If a P^{BIP} can be simplified to either a P^{MK} or P^{SC}, then the following corollary provides a facet-defining condition. This result strengthens Padberg's (1, k) configurations and does require both conditions of a maximal hyperstar.

Theorem 3.6. Let $S_{m,l,k}$ be a maximal induced hyperstar in the conflict hypergraph H_k generated from a multiple knapsack or set covering polytope with $n \geq m \geq k > l+1 \geq 2$. If $k_i \geq \left\lceil \frac{m-l}{2} \right\rceil + k - 1$ for all $i \in L$, and if for each vertex $v \in S_B \cup S_{\bar{B}}$ there exists an edge $d \in E(S_{m,l,k})$

such that $v \in d$ and d is a minimal edge, then

$$\sum_{i\in S_L}(p_i-k+1)x_i-\sum_{i\in \bar{S}_L}(p_i-k+1)x_i+\sum_{i\in S_B}x_i-\sum_{i\in \bar{S}_B}x_i$$
$$=\sum_{i\in S_L}(p_i-k+1)+k-l-1-\left|\bar{S}_B\right|.$$

is a facet-defining inequality.

4. COMPUTATIONAL RESULTS

This section describes how hyperstar inequalities can be used to improve the solution time for the project allocation problem (PA). All computational results reported here were performed on a Pentium IV PC with 1 GB of Ram and a 1.8 GHz processor. The study compares the default setting of CPLEX 7.1 with the default setting of CPLEX 7.1 when a hyperstar inequality is added as a preprocessing cut.

From a set of n projects each with an expected benefit c_i and an expected cost a_i for i = 1, ..., n, the project allocation problem seeks to find the set of projects that maximizes the expected benefit while not spending more than a budget b and choosing at least r projects. The restriction on r projects is included for diversification purposes, which also helps diminish risk. Clearly, PA is *NP*-hard since the knapsack problem is a special case (Karp (1972)).

The project allocation problem is closely related to the knapsack problem and some problems in portfolio management (Bertsimas et al. (1999), Pinto and Rustem (1998)). One such portfolio management problem is the cardinality constrained knapsack problem, which requires that no more than r projects be implemented. Some valid and facet-defining inequalities along with branching techniques for the cardinality constrained knapsack problem can be found in Farias De and Nemhauser (2003).

PA's structure allows us to quickly find hyperstar inequalities. These hyperstar inequalities all had nonbarred hub vertices and barred spoke vertices. The basic idea is that if several high cost projects are selected, then several low cost projects must also be selected in order to guarantee that r total projects can be selected.

To describe these hyperstar inequalities, we may assume, without loss of generality, that a is sorted in descending order ($a_i \le a_j$ for all $i > j$). Now let L = {1, ..., l} where l is some integer. Now select an integer $p \ge 1$ such that $p + 1 < r$ and find the maximum integer q such that $l + 1 \le q \le n - r + l + 1$ and $\sum_{i=1}^{l}a_i+\sum_{i=q}^{q+r-l-p-1}a_i+\sum_{i=n-p+1}^{n}a_i > b$. If no such q exists, then new values for l or p must be obtained. If such a q exists, we claim that {1, ..., l, $\overline{q+r-l-p}$, ..., $\bar{n}$} is a hyperstar with hub {1, ..., l} and blade vertices {$\overline{q+r-l-p}$, ..., $\bar{n}$} in H_k where $k = m - p$ and $m = l + n - q - r + l + p + 1$ or equivalently $k = 2l + n + 1 - q - r$.

To show that the above technique generates a hyperstar we begin by showing that {1, ..., l, $\overline{q+r-l-p}$, ..., $\overline{n-p}$} is an edge in H_k. It suffices to show that $x_1 = ... = x_l = 1$ and $x_{q+r-1-p} = ... = x_{n-p} = 0$ is infeasible. Since a is sorted, the minimum amount of cost that satisfies the above restriction and has r variables equal to 1 occurs when $x_1 = ... = x_l = 1$, $x_{n-p+1} = ... = x_n = 1$ and $x_q = ... = x_{q+r-l-p-1} = 1$. From the preceding paragraph, this exceeds the budget and is infeasible and so it is an edge. Furthermore, due to the sorted order of a all of the other necessary edges are also in H_k and so the induced subhypergraph of these vertices contain a hyperstar. Consequently, Lemma 3.1 provides a valid inequality of the form

$$\sum_{i=1}^{l}(p+1)x_i-\sum_{i=q+r-l-p}^{n}x_i \le (l-1)(p+1).$$

A total of 180 project allocation problems were created by randomly assigning a_i to an integer between 0 and 200,000 and $c_i = a_i$ for i = 1, ..., n. The value of b is $\left\lfloor \frac{\sum_{i=1}^{n}a_i}{5} \right\rfloor$ and r is $\frac{n}{4}$. These instances follow the spirit of both Chvátal (1980) and Hunsaker and Tovey (2004)'s computationally intensive knapsack instances. The aforementioned process was used to create a hyperstar inequality. For this study, the value of l was between 1 and 3 and the value of p was between 3 and 8. In addition, the preprocessing times were exceptionally fast, less than .01 seconds per problem.

The main reason that Chvátal (1980) and Hunsaker and Tovey (2004)'s knapsack instances are difficult to solve is that there exists an optimal solution equal to b with probability approaching 1. Thus, the root relaxation and all

Table 1. Hyperstar inequality computational results

Number Variables	Number Projects	Avg m	Avg $\lvert S_B \rvert$	Avg k	CPLEX Default Time (Sec)	Hyperstar Time (Sec)	Percentage of Improvement
28	7	19.1	17.1	15.1	112	92	17.9
32	8	16.9	14.9	12.9	497	417	16.1
36	9	18.9	16.9	14.8	1189	843	29.1
40	10	29	26	23	2647	1016	62.2
44	11	30.2	28.2	24.2	551	279	49.4
48	12	29.9	27.9	23.9	285	154	46.0
Average		**24.0**	**21.8**	**19.0**	**880**	**467**	**36.7**

other nodes of the branching trees were infeasible, integer or had an LP relaxation of *b*. Thus, the added hyperstar cuts never decreased the root node relaxation. However, the effectiveness of the hyperstar cuts can still be demonstrated by the total time required to solve these problems.

The computational results are shown in Table 1. Each row in this table corresponds to 30 random instances of the specified size. Observe that including hyperstar inequalities decreased the overall run time by about 1/3. In addition, the hyperstars are large and involve the majority of the variables.

Observe that the instances with more variables required less time to solve, which seems counter-intuitive. However, there are so many solutions with a value of *b*, that CPLEX could quickly find an optimal solution equal to *b*. Once CPLEX finds this solution, all branching nodes are immediately fathomed. In contrast, the instances with 28 variables frequently had optimal solutions not equal to *b* and so CPLEX had to explore all LP relaxations with value equal to *b*. With only 28 variables, the number of nodes could still be quickly explored. The middle instances required the longest time since they fell right in the middle of the two. That is, a few instances didn't have an optimal solution equal to *b* and these instances still had relatively large branch-and-cut trees.

5. CONCLUSION AND FUTURE RESEARCH

This paper has presented a new class of valid inequalities, called hyperstar inequalities for general binary integer programs. Some conditions where these inequalities define large dimensional faces are also presented. This paper also introduces a fast method to create useful hyperstar inequalities for the project allocation problem. These hyperstar inequalities decreased the average solution time by approximately one third.

There still remains a substantial amount of research involving hyperstar inequalities and conflict hypergraphs. The two most important research questions are how to efficiently find hyperstar inequalities for general binary integer programs and how useful are hyperstar inequalities.

On top of these open research questions, the results of this paper could be theoretically strengthened by providing conditions when hyperstar inequalities would define facet-defining inequalities. In addition, discovering other subhypergraphs that induce valid inequalities could also be both theoretically and computationally beneficial.

REFERENCES

1. Atamtürk, A., Nemhauser, G.L., and Savelsbergh, M.W.P. (2000). The mixed vertex packing problem. *Mathematical Programming*, 89: 35-53.
2. Atamtürk, A., Nemhauser, G.L., and Savelsbergh, M.W.P. (2000). Conflict graphs in solving integer programming problems. *European Journal of Operational Research*, 121(1): 40-55.
3. Berge, C. (1973). *Graphs and Hypergraphs*, North-Holland Publishing Company, translated by Minieka, E.
4. Bertsimas, D., Darnell, C., and Soucy, R. (1999). Portfolio construction through mixed-integer programming at Grantham, Mayo, Van Otterloo and company. *Interfaces*, 29(1): 49-66.
5. Bixby, R.E. and Lee, E.K. (1994). Solving a truck dispatching scheduling problem using branch and cut. *Operations Research*, 46(3): 355-367.
6. Borndörfer, R. (1997). *Aspects of Set Packing, Partitioning, and Covering*, Ph.D. Thesis, Technischen Universtät Berlin, Berlin, Germany.
7. Cornuejols, G. and Sassano, A. (1989). On the 0,1 facets of the set covering problem. *Mathematical Programming*, 43: 45-55.
8. Chvátal, V. (1973). Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4: 305-337.
9. Chvátal, V. (1980). Hard knapsack problem. *Operations Research*, 28(6): 1402-1412.
10. Easton, T., Hooker, K., and Lee, E. (2003). Facets of the independent set polytope. *Mathematical Programming*, 98: 177-199.
11. Euler, R., Junger, M., and Reinelt, G. (1987). Generalizations of cliques, odd cycles and anticycles and their relation to independence system polyhedra. *Mathematics of Operations Research*, 12: 451-462.
12. Farias De, I. and Nemhauser, G. (2003). A polyhedral study of the cardinality constrained knapsack problem. *Mathematical Programming, Series A*, 96: 439-467.
13. Hooker, K. (2004). *Hypergraphs and Integer Programs*, Ph.D. Dissertation, Industrial and Manufacturing Systems Engineering, Kansas State University.
14. Hunsaker, B. and Tovey, C. (2004). Simple lifted cover inequalities and hard knapsack problems. Technical Report, Industrial Engineering, University of Pittsburgh, Pittsburgh, pp. 1-13.
15. Karp, R. (1972). Reducibility among combinatorial problems. In: R.E. Miller and J.W. Thatcher (Eds.), *Complexity of Computer Computations*, Plenum Press, New York, pp. 85-103.
16. Nemhauser, G. and Trotter, L. (1974). Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6: 48-61.
17. Nemhauser, G. and Wolsey, L. (1988). *Integer and Combinatorial Optimization*, John Wiley and Sons, New York.
18. Padberg, M. (1973). On the facial structure of set packing polyhedra. *Mathematical Programming*, 5: 199-215.
19. Padberg, M. (1980). (1, *k*)-configurations and facets for packing problems. *Mathematical Programming*, 18: 94-99.
20. Pinto, R. and Rustem, B. (1998). Solving a mixed-integermultiobjective bond portfolio model involving logical conditions. *Annals of Operations Research*, 81: 497-513.
21. Pulleyblank, W. and Edmonds, J. (1975). Facets of 1-matching polyhedra. In: C. Berge and D.R. Chaudhuri (Eds.), *Hypergraph Seminar*, Springer,

Heidelberg, pp. 214-242.
22. Sassano, A. (1989). On the facial structure of the set covering polytope. *Mathematical Programming*, 44: 181-202.
23. The CPLEX Solver on ILOG's Home Page. http://www.ilog.com/.