Algorithmic Operations Research Vol.6 (2011) 40–55

Reductions for the Stable Set Problem

E. C. Sewell [a] S. H. Jacobson [b] Hemanshu Kaul [c]

[a]Department of Mathematics and Statistics, Southern Illinois University Edwardsville, Edwardsville, IL 62026-1653
[b]Simulation and Optimization Laboratory, Department of Computer Science, 201 North Goodwin Avenue (MC-258), University of Illinois, Urbana, IL 61801-2302
[c]Department of Applied Mathematics, Illinois Institute of Technology, 10 West 32nd Street, Chicago, IL 60616

Abstract

One approach to finding a maximum stable set (MSS) in a graph is to try to reduce the size of the problem by transforming the problem into an equivalent problem on a smaller graph. This paper introduces several new reductions for the MSS problem, extends several well-known reductions to the maximum weight stable set (MWSS) problem, demonstrates how reductions for the generalized stable set problem can be used in conjunction with probing to produce powerful new reductions for both the MSS and MWSS problems, and shows how hypergraphs can be used to expand the capabilities of clique projections. The effectiveness of these new reduction techniques are illustrated on the DIMACS benchmark graphs, planar graphs, and a set of challenging MSS problems arising from Steiner Triple Systems.

Key words: Maximum stable set; reductions; probing; hypergraphs

1. Introduction

A *stable set* is a set of nodes in a graph that are mutually nonadjacent. The problems of finding a maximum stable set (MSS) or a maximum weight stable set (MWSS) in a graph are NP-complete problems that have received a great deal of attention in the literature [8]. One approach to finding an MWSS is to try to reduce the size of the problem by transforming the problem into an equivalent problem on a smaller graph. Most of the known reductions fall into one of three categories. Inclusion reductions are based on finding a set of nodes S such that there exists an MWSS that includes S. Thus the problem reduces to finding an MWSS in the graph obtained by deleting S and its neighbors. Exclusion reductions are based on finding a set of nodes U such that there exists an MWSS that excludes U, in which case the problem reduces to finding an MWSS in the graph obtained by deleting U. Contraction reductions are based on finding a set of nodes S such that either there is an MWSS that contains S or there is an MWSS that contains the neighbors of S. Thus the problem reduces to finding an MWSS in a graph obtained by replacing S and its neighbors with a single node.

Reductions for the MSS and MWSS problems have been used to study properties of stability critical graphs and facets of the stable set polytope. They have been used algorithmically in heuristics, polynomial time algorithms for special classes of graphs, and exact algorithms. For our purposes, it will be convenient to divide exact algorithms into two types: theoretical and practical. Both types are based on recursive algorithms. Theoretical exact algorithms focus on eliminating redundant branches, without using lower and upper bounds, with the goal of producing a good time bound for the algorithm (e.g., see [7,13,14,19,34,35,42,44,45]). Practical algorithms tend to focus on using lower and upper bounds, in addition to using the techniques used by theoretical algorithms to eliminate redundant branches (e.g., see [2–6,10,12,16,24,25,27,29–33,40,46]). Papers that present theoretical exact algorithms do not usually include any computational results, while papers that present practical exact algorithms do not usually include time bounds (since it is more difficult to compute tight time bounds for these algorithms). Both theoretical and practical exact algorithms have made extensive use of reductions. Details on how reductions have been used are presented in Section 3.

Any binary integer program with two variables per inequality (BIP2VAR) can be written as

$$
\begin{array}{lll}
(BIP2) & z_0 + \max bx & \\
 & x_v \le x_w & \forall\ (v,w) \in A \\
 & x_v + x_w \ge 1 & \forall\ (v,w) \in C
\end{array}
$$

Email: E. C. Sewell [esewell@siue.edu], S. H. Jacobson [shj@illinois.edu], Hemanshu Kaul [kaul@math.iit.edu].

$$x_v + x_w \leq 1 \quad \forall\, (v,w) \in E$$
$$x_v \in \{0,1\} \quad \forall\, v \in V,$$

where z_0 is a constant term (initially zero) that will be used later. This problem has been called the Generalized Stable Set Problem (GSSP) because the MWSS problem is a special case of it with $A = C = \emptyset$. A bigraph is a multigraph that may contain three types of edges: undirected, directed, and bidirected. With every BIP2VAR, there is an associated bigraph B consisting of the set of nodes $V = \{1, ..., n\}$ corresponding to the n variables in BIP2 and the sets of edges A, C, and E. Let $B = (V, A, C, E)$ denote the bigraph and $\alpha_b(B)$ or $\alpha_b(V, A, C, E)$ denote the optimal value of BIP2. If $G = (V, E)$ is a graph, then $\alpha_b(G) \equiv \alpha_b(V, \emptyset, \emptyset, E)$ is the weight of an MWSS in G.

If a bigraph B contains two nodes v and w such that there is more than one type of edge between v and w, then at least one of the nodes can be eliminated. For example, if $(v, w) \in A$ and $(v, w) \in E$, then x_v must be zero in every feasible solution of BIP2, so node v can be eliminated from B; such reductions will be examined in greater detail in Section 2. The purpose of this paper is to introduce new reductions for the stable set problem (Section 3.), show how reductions for the GSSP can be used to generalize reductions for the MSS problem to the MWSS problem (Section 3.), show how probing can be used together with GSSP reductions to achieve greater reductions (Section 4.), and show how clique projections can be extended to hypergraphs to obtain reductions for the stable set problem (Section 5.).

Two problems closely related to the MWSS problem are the maximum weight clique problem and the minimum weight node cover problem. A clique is a set of mutually adjacent nodes, so finding an MWSS is directly equivalent to finding a maximum weight clique in the complement of the graph. A node cover is a set of nodes C such that every edge in E has at least one endpoint in C. If C is a minimum weight node cover, then $V \setminus C$ is an MWSS, so the node cover problem and the MWSS problem also are directly equivalent to each other. Many of the references cited in this paper actually address the maximum clique or minimum node cover problem, but their results can be immediately translated into results for the MWSS problem.

2. Reductions for Generalized Stable Set Problems

The *closure* of a bigraph $B = (V, A, C, E)$ is the bigraph obtained by adding every edge that is implied by the original set of edges. To describe the closure, it will be convenient to assign a plus or minus to the ends of each edge. A plus will be assigned to the ends of each edge in E, a minus will be assigned to the ends of each edge in C, and if $(v, w) \in A$, then a plus will be assigned to the end incident to v and a minus will be assigned to the end incident to w. This assignment of plus and minus signs to an edge (v, w) is obtained by using the sign of x_v and x_w after the corresponding constraint has been written as a less than or equal to inequality with all the variables on the left-hand side of the inequality. Suppose that (u, v) and (v, w) are edges such that (u, v) has a plus assigned to the end incident to v and (v, w) has a minus assigned to the end incident to v. Then adding the two corresponding inequalities (written in less than or equal to form) shows that these two edges imply a third edge, (u, w), where the sign assigned to u will be the same as the sign assigned to u in the edge (u, v) and the sign assigned to w will be the same as the sign assigned to w in the edge (v, w). Johnson and Padberg [20] prove that a bigraph is closed if for every pair of edges, the edge implied by that pair (if any) is already in B. They also develop an $O(n^3)$ algorithm to compute the closure of a bigraph.

It is relatively simple to detect variables that can be eliminated once the closure of B has been computed. The four basic configurations of edges that permit variables to be eliminated are shown in Table 1.

Table 1

Configuration of Edges	Implication	Reduction
1. $(u,v) \in A, (u,v) \in C$	$x_v = 1$	Delete v from B. Add b_v to z_0.
2. $(u,v) \in A, (u,v) \in E$	$x_u = 0$	Delete u from B.
3. $(u,v),(v,u) \in A$	$x_u = x_v$	Delete v from B. Add b_v to b_u.
4. $(u,v) \in C, (u,v) \in E$	$x_u + x_v = 1$	Delete v from B. Add b_v to z_0. Subtract b_v from b_u.

Four Basic GSSP Reductions.

Note that the third reduction transforms an MSS problem into an MWSS problem. Also the fourth reduction can produce a negative value of b_u; to avoid this, choose the node with the smaller objective function coefficient for elimination. If three edges are present between a pair of nodes, then the reductions listed above can be combined. If a pair of nodes has four edges between them, then the problem is infeasible. Infeasibility should not occur in our reductions, because we are always starting with a stable set problem on a graph $G = (V, E)$, which always has a feasible solution. After all possible reductions have been made, there will be at most one edge between any pair of nodes.

For a closed, reduced bigraph, Sewell [39] has shown that the GSSP is actually an MWSS problem in disguise by showing that the constraints in A and C can be ignored.

Theorem 1 *Suppose $B = (V, A, C, E)$ is a closed, reduced bigraph, $G = (V, E)$, and $b \geq 0$. Then $\alpha_b(V, A, C, E) = \alpha_b(V, \emptyset, \emptyset, E)$.*

Now suppose A is a set of constraints of the form $x_u \leq x_v$, C is a set of constraints of the form $x_u + x_v \geq 1$, and E^* is a set of constraints of the form $x_u + x_v \leq 1$. Further suppose that there exists an MWSS in G that satisfies all the constraints in A, C, and E^*. Let $B = (V, A, C, E^* \cup E)$. Then $\alpha_b(G) = \alpha_b(B)$ (the constraints for B contain the constraints for G implies $\alpha_b(G) \geq \alpha_b(B)$; and G contains an MWSS that satisfies the constraints for B implies that $\alpha_b(G) \leq \alpha_b(B)$). Next close and reduce B to obtain $B' = (V', A', C', E')$ with weighting b' and let $G' = (V', E')$. Theorem 1 yields $\alpha_{b'}(B') = \alpha_{b'}(G')$, so if any nodes in B were eliminated during the reduction procedure, then those same nodes can be eliminated from G to produce an equivalent MWSS problem on a smaller graph. That is, reductions in the bigraph B lead directly to reductions in the graph G. This technique will be used to develop several new reductions in Section 3.Methods of obtaining the additional sets of constraints A, C, and E^* via probing will be discussed in Section 4.

3. Direct Reductions for the Stable Set Problem

In this section we review many of the reductions that have been used for the MSS problem and note how they have been used in the literature. We also extend several of these reductions to the weighted case and introduce a number of new reductions. Throughout this section, let $G = (V, E)$ be a graph and $b \geq 0$ be a nonnegative weighting of the nodes of G.

A few definitions are necessary before proceeding. The *neighborhood of node* u in G is defined as $N(u) = \{v \in V : (u, v) \in E\}$ and the *neighborhood of a set of nodes* U is defined as $N(U) = \{v \in V \backslash U : \exists u \in U \ni (u, v) \in E\}$. $N^2(u)$ is defined to be $N(N(u)) \backslash (\{u\} \cup N(u))$. The *degree of node* v is $d(v) = |N(v)|$. If $V' \subseteq V$, then $b(V') = \sum_{v \in V'} b_v$ and $G[V']$ denotes the subgraph of G induced by V'. For $v \in V$, $G - v$ will be used to denote $G[V \backslash \{v\}]$ and $G - V'$ will denote $G[V \backslash V']$ for $V' \subseteq V$.

3.1. *Exclusion Reductions*

Exclusion reductions are based on finding a set of nodes U such that there exists an MWSS that excludes U. In this case, the problem reduces to finding an MWSS in the graph obtained by deleting U. A node v in graph G *dominates* a node u if $(u, v) \in E$ and $N(v) \subseteq \{u\} \cup N(u)$. For the MSS problem, if there is an MSS that contains u, then there is also an MSS that contains v but not u. Hence there is an MSS that excludes u, which means that u can be deleted from the graph. This reduction has been used in several of the fastest-known theoretical exact algorithms for the MSS problem [7,34,35,44]. This reduction can be generalized to the MWSS problem as shown in the following proposition.

Proposition 2 *If a node v dominates a node u in a graph $G = (V, E)$ and $b_v \geq b_u$, then $\alpha_b(G - u) = \alpha_b(G)$ (i.e., node u can be deleted from G).*

Proof. Let S be an MWSS. If S contains u, then S cannot contain v, so $S' = (S \backslash \{u\}) \cup \{v\}$ is a stable set with $b(S') \geq b(S)$, and hence S' is an MWSS that excludes u. Thus $\alpha_b(G - u) = \alpha_b(G)$. ■

Another exclusion reduction can be defined for a closed, reduced bigraph B. If a node u is connected to every other node in B by either an edge in A or an edge in E, then the weight of the heaviest stable set that contains u equals $b_u + \sum_{v:(u,v) \in A} b_v$. In such a case, the search for a heavier stable set can be restricted to $B - u$, hence the bigraph can be reduced by excluding u.

3.2. *Inclusion Reductions*

Inclusion reductions are based on finding a set of nodes S such that there exists an MWSS that includes S. Thus the problem reduces to finding an MWSS in the graph obtained by deleting S and its neighbors.

One type of inclusion reduction is based on simplicial nodes. A node u in graph G is *simplicial* if $N(u)$ is a clique. For the MSS problem, it is easy to see that if u is simplicial, then there must be an MSS that contains u. Reductions based on simplicial nodes have been used in perfect elimination schemes for chordal graphs (see [37] for a discussion of chordal graphs and further references). Mannino and Sassano [24] report that simplicial reductions were crucial in solving some of the larger MSS problems in the DIMACS test set. Special cases of simplicial nodes, that have been used in several of the theoretical exact algorithms for the MSS problem, are nodes that have degree zero, one, or two (as part of a triangle) [34,35,44]. The situation is slightly more complicated for the MWSS problem, since there may not be an MWSS that includes u. Nonetheless, a reduction can be made, as given in the next proposition.

Proposition 3 *Suppose u is a simplicial node in a graph $G = (V, E)$ with node weights $b \geq 0$. Let*

$$b'_v = \begin{cases} 0 & \text{if } v = u \text{ or } b_v \leq b_u \text{ and } v \in N(u) \\ b_v - b_u & \text{if } b_v > b_u \text{ and } v \in N(u) \\ b_v & \text{otherwise} \end{cases}$$

and let G' be the graph obtained by deleting all nodes $v \ni b'_v = 0$. Then $\alpha_{b'}(G') = \alpha_b(G) - b_u$.

Proof. If $v \in N(u)$ and $b_v \leq b_u$, then v can be deleted by Proposition 2 since u dominates v. So we can now assume $b_v > b_u\ \forall v \in N(u)$. Every MWSS must contain exactly one node of $\{u\} \cup N(u)$, so subtracting b_u from each node in $\{u\} \cup N(u)$ decreases $\alpha_b(G)$ by precisely b_u. ■

Another type of inclusion reduction is based on the surplus function. For $V' \subseteq V$, let $\Gamma(G, V') = \{w \in V : \exists v \in V' \ni (v, w) \in E\}$. Hence $\Gamma(V') = N(V')$ if and only if V' is a stable set. Lovász and Plummer [23] introduced a *surplus* function $\sigma(G, V') = |\Gamma(V')| - |V'|$ which Sewell [38] generalized to the weighted case as $\sigma_b(G, V') = b(\Gamma(G, V')) - b(V')$. Whenever G is clear from the context, it will be suppressed from the notation of Γ and σ_b. Furthermore, the subscript b will be suppressed from $\sigma_b(V')$ if $b_v = 1\ \forall v \in V$. The next theorem gives sufficient conditions, in terms of the surplus function, under which a stable set will be contained in every MWSS [18].

Theorem 4 (Hammer, Hansen, and Simeone) *Suppose $b > 0$, $\sigma_b(I) = \min\{\sigma_b(I') : I'$ is a stable set in $G\}$, and I has minimum cardinality among all such minimizers (note that I may be empty). Then I is contained in every MWSS.*

By Theorem 4, G can be reduced by eliminating I and its neighbors. Let $G' = G[V \backslash (I \cup N(I))]$. By the definition of I, it can be shown that $\sigma_b(G', S) \geq 0$ for every stable set S in G'. In [38] it was shown that if there exists a stable set S in G' such that $\sigma_b(S) = 0$, then there exists an MWSS containing S. Consequently, G' can be further reduced by eliminating $S \cup N(S)$. The sets I and S can be found in polynomial time - see [11,18,28] for details. These reductions have played a pivotal role in fixed-parameter algorithms for the vertex cover problem [13,14], which in turn have yielded several of the fastest-known theoretical exact algorithms for the MSS problem.

A third type of inclusion reduction is based on the following theorem [28].

Theorem 5 (Nemhauser and Trotter) *If S is an MWSS in $G[S \cup N(S)]$, then S is contained in an MWSS in G.*

This reduction technique includes the simplicial reduction for the MSS problem (but not for the MWSS problem) since if u is simplicial, then it is an MSS in $\{u\} \cup N(u)$. This reduction also includes the surplus reductions. The following proposition is needed to prove this.

Proposition 6 *Suppose $G = (V, E)$ is a graph with node weights $b \geq 0$ and S is a nonempty stable set such that $\sigma_b(S) \leq 0$. If S is not an MWSS in $G[S \cup N(S)]$, then there exists a subset S' of S such that $\sigma_b(S') < \sigma_b(S)$.*

Proof. Let I be an MWSS in $G[S \cup N(S)]$. By assumption, S is not an MWSS in G, hence $b(I) > b(S)$. Let $I_S = I \cap S$ and $I_N = I \cap N(S)$. Clearly, $I_N \neq \emptyset$, otherwise S would be an MWSS in $G[S \cup N(S)]$. If $I \subseteq N(S)$, then $\sigma_b(S) = b(N(S)) - b(S) \geq b(I) - b(S) > 0$, which contradicts that $\sigma_b(S) \leq 0$. Hence $I_S \neq \emptyset$. Then

$$\begin{aligned} \sigma_b(I_S) &= b(N(I_S)) - b(I_S) \\ &\leq b(N(S) \setminus I_N) - b(I_S) \\ &= b(N(S)) - b(I_N) - b(I_S) \\ &= b(N(S)) - b(I_N \cup I_S) \\ &= b(N(S)) - b(I) \\ &< b(N(S)) - b(S) \\ &= \sigma_b(S). \end{aligned}$$

The first inequality holds because $N(I_S) \subseteq N(S) \backslash I_N$. The second inequality holds because $b(I) > b(S)$. Therefore, I_S is a subset of S such that $\sigma_b(I_S) < \sigma_b(S)$. ■

Let I be as defined in Theorem 4 and S be as defined immediately after Theorem 4. Proposition 6 implies that I is an MWSS in $I \cup N(I)$ because I is defined to be the stable set with the smallest surplus. Similarly, Proposition 6 implies that S is an MWSS in $G'[S \cup N(S)]$ because $\sigma_b(G', S) = 0$ and every stable set in G' has nonegative surplus. Thus, Theorem 5 includes the surplus reductions. In general, it is difficult (i.e., NP-complete) to find a set S that is an MWSS in $G[S \cup N(S)]$, whereas all the other reductions discussed so far in this section can be found in polynomial time.

3.3. *Contraction Reductions*

Let the *contraction of* $V' \subseteq V$ be defined as the graph obtained from G by replacing V' with a single node that is connected to every node in $N(V')$. If it is possible to predict how the contraction changes the optimal value of the MWSS problem, then the problem of finding an MWSS in G can be reduced to finding an MWSS in the contraction of V'. The best-known contraction for the MSS problem works on nodes of degree two. Suppose $d(u) = 2$ and the two neighbors of u, say v and w, are not adjacent (if they are adjacent, then u is simplicial). Suppose also that S is an MSS. If S contains exactly one of v, w, then it can be transformed into an MSS that includes u. Thus, either there is an MSS that contains u or an MSS that contains $N(u)$. The *contraction of node* u is defined as the graph G' obtained from G by replacing u and $N(u)$ with a single node that is connected to every node in $N^2(u)$. It is easy to see that $\alpha(G') = \alpha(G) - 1$. The contraction of nodes of degree two has been used in many theoretical exact algorithms (e.g., [7,14,44]). This contraction has also played a fundamental role in analyzing the structure of stability critical graphs and facets of the stable set polytope (e.g., [21–23,38,41]). This contraction is a special case of the struction operation defined in [1,15].

The contraction of a node of degree two can be generalized in two different ways. First, this type of contraction can be extended to the weighted case, as long as the weight of the node of degree two is greater than or equal to the weight of at least one of its neighbors.

Proposition 7 *Suppose u is a node of degree two, $N(u) = \{v, w\}$ are the neighbors of u, $(v, w) \notin E$, and $b_v \leq b_w$.*

(1) If $b_v \leq b_u \leq b_w$, then $\alpha_{b'}(G') = \alpha_b(G) - b_u$, where G' is the graph obtained from G by deleting u, connecting v to every node in $N(w)$, and letting $b' = b$, except $b'_w = b_w - b_u$.

(2) If $b_w < b_u < b_v + b_w$, then $\alpha_{b'}(G') = \alpha_b(G) - b_u$, where G' is the graph obtained from G by deleting u and w, connecting v to every node in $N(w)$, and letting $b' = b$, except $b'_v = b_v - (b_u - b_w)$.

(3) If $b_v + b_w \leq b_u$, then $\alpha_b(G') = \alpha_b(G) - b_u$, where G' is the graph obtained from G by deleting u, v, and w.

Proof.

(1) Because $b_u \geq b_v$, then there exists an MWSS S such that either S contains u or w. Hence the inequality $x_u + x_w \geq 1$ can be added to BIP2 without changing its optimal value. Let $C = \{(u, w)\}$ and $B = (V, \emptyset, C, E)$. First, close B, which results in adding all possible edges between v and $N(w)$. Next apply Reduction (4) from Table 1, which deletes u and reduces both the optimal value and b_w by b_u.

(2) First, G can be reduced in the same manner as described in case (1), except node w rather than node u is deleted, because $b_w < b_u$. Node u has degree one in the resulting graph, so it is simplicial. Consequently, Proposition 3 can be applied to delete u and to reduce the weight of v by $b_u - b_w$.

(3) In this case, u is an MWSS in $\{u\} \cup N(u)$, so Theorem 5 implies that u is contained in an MWSS. Hence u and its neighbors can be deleted from G.

■

Note that if $b_u = b_v = b_w = 1$, then the reduction specified by Proposition 7(1) is precisely the same as the contraction of node u. The proof of Proposition 7(1) demonstrates that contracting a node of degree two can be viewed as a GSSP reduction.

The second way that the contraction of a node of degree two can be generalized is to notice that $\sigma(u) = 1$ whenever u is a node of degree two. We need the following theorem to achieve this generalization.

Theorem 8 *Suppose S is a stable set such that $\sigma_b(S) = k$ for some $k \geq 0$ and $\sigma_b(S') \geq k$ for all nonempty subsets $S' \subseteq S$. Then there exists an MWSS I such that either $S \subseteq I$ or $I \cap S = \emptyset$.*

Proof. If $I \cap S = \emptyset$, then there is nothing to prove, so let I be an MWSS such that $I \cap S \neq \emptyset$. Let $I_S = I \cap S$ and $I_N = I \cap N(S)$. Then

$$\begin{aligned} b(I \cap (S \cup N(S))) &= b(I_S) + b(I_N) \\ &\leq b(N(I_S)) - k + b(I_N) \\ &\leq b(N(S)) - k \\ &= b(S). \end{aligned}$$

The first inequality holds because $\sigma_b(I_S) \geq k$ implies $b(N(I_S)) - b(I_S) \geq k$. The second inequality holds because $I_N \subseteq N(S)$ and $I_N \cap N(I_S) = \emptyset$ (since I is a stable set). Therefore, $S \cup I \backslash I_N$ is an MWSS that contains S. ■

Theorem 8 yields another useful reduction. Recall that the contraction of $V' \subseteq V$ is defined as the graph obtained from G by replacing V' with a single node that is connected to every node in $N(V')$.

Corollary 9 *Suppose S is a stable set such that $\sigma_b(S) = k$ for some $k \geq 0$ and $\sigma_b(S') \geq k$ for all nonempty subsets $S' \subseteq S$. Let G' be the graph obtained by contracting S to a single node, say s, and let $b_s = b(S)$. Then $\alpha_b(G') = \alpha_b(G)$.*

Proof. The proof begins by showing that $\alpha_b(G) \geq \alpha_b(G')$. Every MWSS in G' can be transformed into a stable set in G of equal weight. To see this, let I' be an MWSS in G'. If $s \in I'$, then $I = S \cup I' \backslash \{s\}$ is a stable set in G and $b(I) = b(I')$. If $s \notin I'$, then I' is stable in G. In either case, I' has been transformed into a stable I in G with weight $b(I)$, hence $\alpha_b(G) \geq \alpha_b(G')$. Conversely, let I be an MWSS in G. Theorem 8 implies that it can be assumed that either $S \subseteq I$ or $I \cap S = \emptyset$. If $S \subseteq I$, then $I' = \{s\} \cup I \backslash S$ is a stable set in G' and $b(I') = b(I)$. If $I \cap S = \emptyset$, then I is a stable set in G'. In either case, I has been transformed into a stable set in G' with weight $b(I)$, hence $\alpha_b(G) \leq \alpha_b(G')$. Therefore $\alpha_b(G') = \alpha_b(G)$. ■

Theorem 8 is used now to prove a reduction that generalizes the contraction of a node of degree two.

Theorem 10 *Suppose S is a stable set such that $\sigma_b(S) = k$, where $k = \min_{v \in N(S)} b_v$, and $\sigma(S') \geq k$ for all nonempty subsets $S' \subseteq S$.*

(1) If $N(S)$ is stable, then $\alpha_b(G') = \alpha_b(G) - b(S)$, where G' is the graph obtained by contracting $S \cup N(S)$ to a single node, say s, and $b_s = k$.

(2) If $N(S)$ is not stable, then $\alpha_b(G') = \alpha_b(G) - b(S)$, where G' is the graph obtained by deleting $S \cup N(S)$.

Proof. We want to show that there exists an MWSS I such that either $S \subseteq I$ or $N(S) \subseteq I$. According to Theorem 8, there exists an MWSS I such that either $S \subseteq I$ or $I \cap S = \emptyset$. If $b(I \cap (S \cup N(S))) \leq b(S)$, then I can be transformed into an MWSS that contains S. Furthermore, the proof of Theorem 8 demonstrates that if $I \cap S \neq \emptyset$, then $b(I \cap (S \cup N(S))) \leq b(S)$. Thus, we can assume without loss of generality that $S \subseteq I$ if and only if $b(I \cap (S \cup N(S))) \leq b(S)$. But $\sigma_b(S) = b(N(S)) - b(S) = k$ implies $b(N(S)) = b(S) + k$. Since $k = \min_{v \in N(S)} b_v$, then the only way that $b(I \cap (S \cup N(S))) > b(S)$ can occur is if $N(S) \subseteq I$. Therefore, either $S \subseteq I$ or $N(S) \subseteq I$. Clearly, if $N(S)$ is not stable, then I cannot contain $N(S)$, and hence it must contain S. In this case, S is contained in an MWSS, thus justifying the reduction in case (2). If $N(S)$ is stable, then the proof of case (1) is completed in a manner similar to the proof of Corollary 9. ■

For the MSS problem, if S is a stable set, then $\sigma(S) = |N(S)| - |S|$, so Theorem 10 implies that if S is a stable set such that $\sigma(S) = 1$ and $\sigma(S') \geq 1$ for all nonempty subsets $S' \subseteq S$, then either $S \cup N(S)$ can be contracted (if $N(S)$ is stable) or $S \cup N(S)$ can be deleted (if $N(S)$ is not stable). This generalizes the case of a node of degree two. Theorem 4 and the discussion following it describe how reductions can be performed to ensure that $\sigma(S) > 0$ (i.e., $\sigma(S) \geq 1$) for every nonempty stable set S. This implies that the condition $\sigma(S') \geq 1$ for all nonempty subsets S' of S is satisfied. Theorem 10 can then be used to perform reductions that ensure $\sigma(S) > 1$ (i.e., $\sigma(S) \geq 2$) for every nonempty stable set S.

4. Probing

Probing is a method that attempts to find relationships between binary variables by temporarily fixing one variable to either zero or one. In this section we show how to obtain stronger reductions for the MWSS problem by combining probing techniques together with the reductions given in Section 3. and reductions for GSSP. The basic idea is that we probe on a node, say u, by trying to put it in an MWSS or by trying to exclude it from an MWSS. We then use the reductions from Section 3. to derive additional binary constraints between u and other nodes in the graph. These additional binary constraints are added to BIP2, which can then be closed and reduced to (possibly) yield stonger reductions for the original graph. Throughout this section, let $G = (V, E)$ be a graph and $b \geq 0$ be a nonnegative weighting of the nodes of G.

As a simple example, suppose $(u, v) \in E$ and u is simplicial in $G - v$. Such a node will be called *nearly-simplicial.* If v is not in any MSS, then every MSS must be contained in $G - v$. But there exists an MSS in $G - v$ that contains u, since u is simplicial in $G - v$. Thus, the constraint $x_u + x_v \geq 1$ can be added to BIP2 without changing the optimal value. The corresponding bigraph can be closed and reduced (using Reduction (4) from Table 1, which deletes u and v (and any nodes adjacent to both u and v). Notice that these reductions cannot be obtained directly from any of the reductions given in Section 3.

The reductions given in Section 3 will be referred to as *direct reductions.* The In-Probe algorithm is given in Figure 1. The parameters of the algorithm are a closed, reduced bigraph $B = (V, A, C, E)$ (together with its integer programming representation BIP2), a set R of direct reductions for the MSS or MWSS problem, and a node u on which to probe. The algorithm begins by temporarily fixing $x_u = 1$, temporarily fixing any other variables in BIP2 that must be zero whenever x_u is one, and temporarily fixing any other variables that must be one whenever x_u is one. The direct reductions in R are then applied to the free variables (i.e., not temporarily fixed). If the direct reductions find a variable x_v that can be set to zero, then the inequality $x_u + x_v \leq 1$ can be added to BIP2. Similarly, if the direct reductions find a variable x_v that can be set to one, then the inequality $x_u \leq x_v$ can be added to BIP2. Finally, if any inequalities have been added to BIP2, then B can be closed and reduced, possibly resulting in a reduction for the original problem.

In-Probe(B, R, u)
// $B = (V, A, C, E)$ is a closed reduced bigraph (BIP2 corresponds to B)
// R is a set of direct reduction techniques
// $u \in V$
$x_u = 1$
$x_v = 0\ \forall v \ni (u, v) \in E$
$x_v = 1\ \forall v \ni (u, v) \in A$
Let V' be the set of nodes in V that have not been temporarily fixed to 0 or 1
Apply the reductions in R to the bigraph induced by V'
Let V'_0 be the set of nodes fixed to 0 by the reductions in R
Let V'_1 be the set of nodes fixed to 1 by the reductions in R
For all $v \in V'_0$ add (u, v) to E
For all $v \in V'_1$ add (u, v) to A
Close and reduce $B = (V, A, C, E)$

Fig. 1. The In-Probe Algorithm.

Example 11 *Consider a graph which contains the subgraph induced by $\{t\} \cup N(t) \cup N^2(t)$ shown in Figure 2. If an in-probe is performed on node t, then u and w will be in an MSS in $G - t - N(t)$, because the stable set $\{u, w\}$ satisfies the conditions of Theorem 10(2). Hence y_i, $i = 1, \ldots, 5$, will be excluded, which means that (t, y_i) can be added to E for $i = 1, \ldots, 5$. After these edges have been added, t is dominated by v, so t can be eliminated. Now an in-probe on v implies that u is in an MSS in $G - t - v - N(v)$, because it is simplicial. This in turn implies that w can also be included in the MSS in $G - t - v - N(v)$, since it has degree one. Hence the in-probe on v adds (v, u) and (v, w) to A. An in-probe on u adds (u, v) to A, and an in-probe on w adds (w, v) to A. Reduction (3) from Table 1 can be applied (after closing the bigraph), which identifies nodes u, v, and w, with weight three. The resulting subgraph is shown in Figure 3. Theorem 10(2) implies that there is an MSS that contains v if y_1, y_4, or y_5 is deleted, hence (v, y_1), (v, y_4), and (v, y_5) can be added to C. When the resulting bigraph is closed and reduced, the only nodes from the subgraph that will remain are y_2 and y_3, with all possible edges added between $\{y_2\} \cup \{y_3\}$ and $N(y_1) \cup N(y_4) \cup N(y_5)$. The weights of all the nodes will be one.*

The Out-Probe algorithm, shown in Figure 4, is similar to the In-Probe algorithm, except that the node v on which we probe has x_v temporarily fixed to zero.

Out-Probe(B, R, v)
// $B = (V, A, C, E)$ is a closed reduced bigraph (BIP2 corresponds to B)
// R is a set of direct reduction techniques
// $v \in V$
$x_v = 0$
$x_u = 0\ \forall u \ni (u, v) \in A$
$x_u = 1\ \forall u \ni (u, v) \in C$
Let V' be the set of nodes in V that have not been temporarily fixed to 0 or 1
Apply the reductions in R to the bigraph induced by V'
Let V'_0 be the set of nodes fixed to 0 by the reductions in R
Let V'_1 be the set of nodes fixed to 1 by the reductions in R
For all $u \in V'_0$ add (u, v) to A
For all $u \in V'_1$ add (u, v) to C
Close and reduce $B = (V, A, C, E)$

Fig. 4. The Out-Probe Algorithm.

Example 12 *Consider a graph which contains the subgraph induced by $\{t\} \cup N(t) \cup N^2(t)$ shown in Figure 5. If an out-probe is performed on node t, then u, v, and w will all be in an MSS in $G - t$, because they are simplicial in $G - t$. Consequently, (t, u), (t, v), and (t, w) can all be added to C. When the bigraph is closed and reduced, the subgraph in Figure 5 will be replaced by the subgraph in Figure 6, where v now has weight $b_v = 2$.*

It is important to distinguish between direct reductions that directly set a variable to zero or one, such as the first two reductions given in Table 1, and those that do not directly set a variable to zero or one, such as the last two reductions given in Table 1 and the contraction reductions. Reductions of the latter type essentially delay the decision about the value of a variable by using a substitution of variables. For example, Reduction (3) from Table 1 uses the substitution $x_u = x_v$ to delay the decision regarding the value of x_v. Similarly, the contraction of a node of degree two, say node u with neighbors v and w, delays the decision about whether u is in the MSS or both v and w are in it. For purposes of probing, reductions of both type may be used during

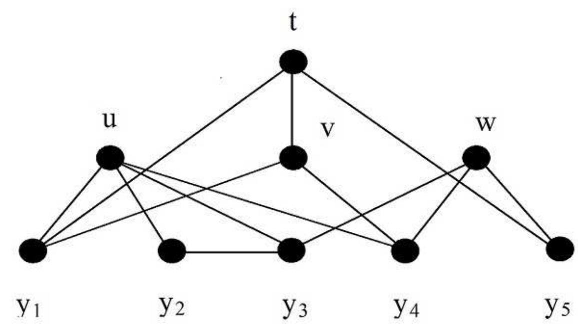

Fig. 2. Induced subgraph containing $t \cup N(t) \cup N^2(t)$.

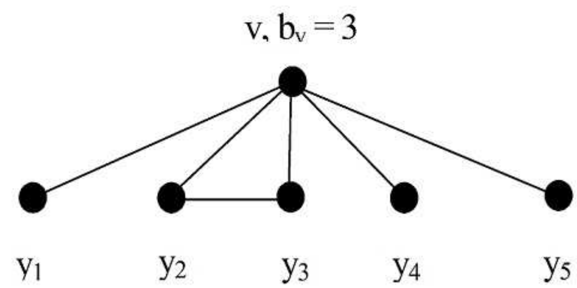

Fig. 3. The subgraph from Figure 2 after performing in-probes on nodes $t, v, u,$ and w.

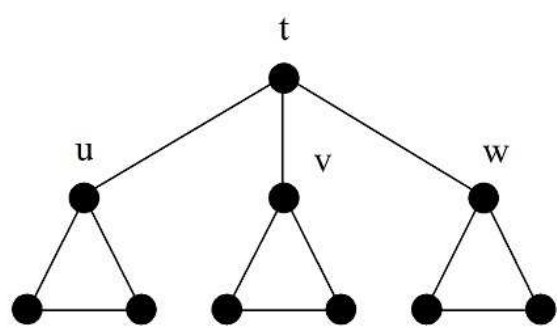

Fig. 5. Subgraph induced by $\{t\} \bigcup N(t) \bigcup N^2(t)$.

the probe, but V_0' and V_1' must include only variables that have been fixed at zero or one. In particular, nodes eliminated by the following reductions should not be included in V_0' and V_1': Proposition 3 regarding weighted simplicial nodes, contraction of nodes of degree two for the MSS problem, parts (1) and (2) of Proposition 7 regarding contraction of weighted nodes of degree two, Corollary 9 regarding the contraction of a stable set, and part (1) of Theorem 10 regarding the contraction of a stable set and its neighbors.

4.1. *Computational Results*

Computational experiments for MSS problems were executed on a 2.0 GHz, dual core, Intel T7200 processor with 3.25 GB of memory. The algorithm was implemented in the C++ programming language. The code was not parallelized, so it only utilized one of the two

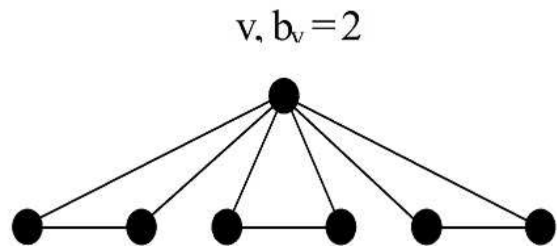

Fig. 6. Subgraph from Figure 5 after performing an out-probe on node t.

cores. The algorithm, called the Reduction-Probe Algorithm (RPA), repeatedly searches for direct reductions until it is no longer able to add any new edges or fix any additional nodes. After the direct reductions, it performs an in-probe on each node and then performs an out-probe on each node. Probing is repeated until it is no longer able to add any new edges or fix any additional nodes.

The power of probing reductions is illustrated on the DIMACS Benchmark graphs, which were collected for the Second DIMACS Implementation Challenge on Clique, Graph Coloring, and Satisfiability [1] . RPA was run on all the Dimacs Benchmark graphs; Table 2 presents the results for the graphs for which RPA was able to fix any variables or add any new edges. In Table 2, Nodes is the number of nodes in the original graph, Fixed is the number of variables fixed, New Edges is the number of new edges found, and CPU is the running time, in seconds.

Table 2 compares RPA to PrePro, which is a preprocessing algorithm for unconstrained quadratic binary optimization (QUBO) problems that was developed by Boros, Hammer, and Tavares [9]. They formulated the MSS problem as a QUBO and then applied PrePro to the DIMACS Benchmark graphs. Table 2 presents the results for the graphs for which PrePro was able to fix any variables or add any new edges. For PrePro, the number of New Edges reported in Table 2 is the number of new edges in the graph after all fixed variables have been removed, whereas New Edges for RPA equals all the new edges found for the original graph. Therefore, these two columns are only directly comparable for graphs where no variables were fixed by PrePro.

Both RPA and PrePro were able to fix all the variables for the c-fat graphs and the hammingx-2 graphs. RPA was able to fix some of the variables for the four mann graphs, whereas PrePro was unable to do so. Furthermore, RPA was able to add some edges to nine other graphs for which PrePro was unable to do so. In terms of execution times, RPA was run on a 2.0 GHz processor while PrePro was run on a 2.8 GHz processor. RPA was faster than PrePro for all the graphs. For the c-fat500 graphs, it was roughly two orders of magnitue faster. In comparing the performance of RPA to PrePro, it should be kept in mind that RPA was designed specifically for the MSS problem while PrePro was designed for the more general QUBO problem, which includes the MSS problem as a special case. Therefore, it is somewhat like comparing apples to oranges, and it is not surprising that RPA is able to find more reductions and that it requires less CPU.

Boros et al [9] also reported the results of applying PrePro to a series of planar graphs that were generated by the LEDA software package. Table 3 compares the performance of RPA to PrePro on these graphs. Each line in the table presents the average for 100 graphs. Both algorithms were able to fix all the variables for all the graphs. RPA is roughly 20 to 40 times faster than PrePro on these graphs.

Table 3

	RPA			**PrePro**	
Nodes	**Fixed**	**New Edges**	**CPU**	**Fixed**	**CPU**
1,000	1,000	10.63	0.0012	1,000	0.05
2,000	2,000	29.60	0.0069	2,000	0.16
3,000	3,000	47.77	0.0085	3,000	0.27
4,000	4,000	108.30	0.0220	4,000	0.53

LEDA planar graphs.

The order in which the probing is performed can make a difference in the number of variables that are fixed and the number of edges that are added. The re-

[1] ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique

Table 2

		RPA			PrePro		
			New*			New*	
Graph	**Nodes**	**Fixed**	**Edges**	**CPU**	**Fixed**	**Edges**	**CPU**
c-fat200-1	200	200	21,433	0.20	200	0	21.1
c-fat200-2	200	200	23,134	0.14	200	0	17.4
c-fat200-5	200	200	28,372	0.13	200	0	3.7
c-fat500-1	500	500	129,208	2.17	500	0	677.0
c-fat500-2	500	500	133,888	2.16	500	0	377.1
c-fat500-5	500	500	147,940	2.16	500	0	166.2
c-fat500-10	500	500	171,376	2.05	500	0	89.5
hamming6-2	64	64	0	0.00	64	0	0.0
hamming8-2	256	256	0	0.00	256	0	0.0
hamming10-2	1,024	1,024	0	0.00	1,024	0	0.1
hamming6-4	64	0	224	0.05	0	0	0.1
hamming8-4	256	0	0	1.38	0	0	8.9
hamming10-4	1,024	0	0	84.88	0	0	265.4
mann_a9	45	14	192	0.01	-	-	-
mann_a27	378	76	3,460	1.55	-	-	-
mann_a45	1,035	184	15,319	16.88	-	-	-
mann_a81	3,321	583	87,828	285.45	-	-	-
p_hat300-1	300	0	74	10.98	-	-	-
p_hat300-2	300	0	31	5.99	-	-	-
p_hat500-1	500	0	6	36.25	-	-	-
p_hat500-2	500	0	4	30.19	-	-	-
p_hat700-1	700	0	1	146.17	-	-	-
san200_0.7_2	200	0	66	1.08	-	-	-
san400_0.5_1	400	0	2,776	21.63	-	-	-
san1000	1,000	0	1,989	422.17	-	-	-

*DIMACS Benchmark Problems. *New Edges for the RPA equals all the new edges found for the original graph. New Edges for PrePro equals the number of new edges in the graph after all fixed variables have been removed. A dash indicates that [9] did not present results because PrePro was unable to fix any variables or add any new edges.*

sults presented above are based on performing the in-probes before the out-probes. For the DIMACS Benchmark graphs, the results were identical except for the mann graphs. For the mann graphs, substantially more variables were fixed by performing the out-probes before the in-probes. As shown in Table 4, RPA was able to reduce the number of nodes by approximately 33% on these challenging problems. It should be noted that these graphs cannot be reduced by any of the direct reduction techniques given in Section 3.

Table 4

Graph	Nodes	Fixed	New Edges	CPU
mann_a9	45	14	192	0.2
mann_a27	378	121	6,084	1.53
mann_a45	1,035	332	29,040	18.30
mann_a81	3,321	1,088	172,800	338.47

RPA with out-probes before in-probes applied to Mann graphs.

5. Clique Projections and Hypergraphs

The reductions in the previous sections all transformed a stable set problem on a graph into another stable set problem on a smaller graph in such a way that the problem could still be formulated as a binary integer program with two variables per inequality. In this section we extend a reduction technique, called clique projection, in such a way that it creates constraints with more than two variables per inequality, and illustrate the power of this reduction technique on the Steiner Triples Systems graphs. Throughout this section, let $G = (V, E)$ be a graph and $b \geq 0$ be a nonnegative weighting of the nodes of G.

A short review of clique projections is first presented. Lovász and Plummer [23] introduced reducible cliques as part of a polynomial time algorithm to find an MSS in a claw-free graph. They defined a maximal clique K contained in G to be *reducible* if $\alpha(G[N(K)]) \leq 2$, and reduced G by letting G' be the graph obtained by deleting the nodes in K and adding an edge (if not already present) between every pair of nodes u and v such that $K \subseteq N(u) \cup N(v)$. If K is reducible, then $\alpha(G') = \alpha(G) - 1$. De Simone and Sassano [43] used an extension of this reduction, which was developed by Sassano [36], to create a polynomial-time algorithm to find an MSS in a bull-free chair-free graph.

Mannino and Sassano [25] introduced edge projections as a specialization of Lovász and Plummer's [23] reduction. Let $e = (u, v) \in E$. Define $N_{uv} = N(u) \cap N(v)$, $N_u = N(u) - N_{uv} - \{v\}$, and $N_v = N(v) - N_{uv} - \{u\}$. The *edge projection of* e is the graph obtained from G by deleting $\{u\} \cup \{v\} \cup N_{uv}$ and adding edges to ensure that every node in N_u is adjacent to every node in N_v. In the case that u is a node of degree two and is not simplicial, the edge projection of e creates the same graph as contracting u (as defined in the first paragraph of Section 3.3.). They also developed an upper bound for $\alpha(G)$ based on edge projections, and incorporated the upper bound into a branch and bound algorithm to produce a fast, practical exact algorithm for the MSS problem.

Mannino and Stefanutti [26] generalized edge projections to the weighted case as follows. Let $e = (u, v) \in E$ and $b_e = \min(b_u, b_v)$. The *weighted edge projection of* e is the graph obtained from G by subtracting b_e from both b_u and b_v, deleting the nodes in N_{uv}, deleting u if its new weight is zero, deleting v if its new weight is zero, and adding edges to ensure that every node in N_u is adjacent to every node in N_v. They created a heuristic for the MWSS problem by embedding a sequence of edge projections in a tabu search.

Both the unweighted and weighted edge projections are special cases of closing and reducing a bigraph. From the graph $G = (V, E)$ create a bigraph $B = (V, \emptyset, C, E)$, where $C = \{e\} = \{(u, v)\}$ is the edge to be projected. Next close and reduce B to obtain $B' = (V', A', C', E')$ with weighting b' and let $G' = (V', E')$. It is straightforward to show that G' is precisely the same as the edge projection of e.

A reduction technique that generalizes reducible cliques and edge projections is now presented. This technique creates constraints with more than two variables per inequality. The new constraints can be conveniently represented by edges in a hypergraph. Given a set of nodes V, a *hyperedge* is a subset of V and the *hyperedge inequality* for hyperedge h is $\sum_{v \in h} x_v \leq |h| - 1$. The definition of a stable set can be extended to a hypergraph $H = (V, E_H)$ as a solution to the integer program:

$$
BIP(H, b) \quad \begin{aligned} \alpha_b(H) &= \max bx \\ \sum_{v \in h} x_v &\leq |h| - 1 \quad \forall\, h \in E_H \\ x_v &\in \{0, 1\} \qquad \forall\, v \in V. \end{aligned}
$$

If $|h| = 2$ for every hyperedge $h \in E_H$, then H is the same as the ordinary graph $G = (V, E_H)$ and BIP(H, b) is the same as BIP2 with $A = C = \emptyset$. If the incidence vector of $S \subseteq V$ is a feasible solution of BIP(H, b), then S will be called a stable set of H and will be said to be a feasible solution of BIP(H, b). Given a hypergraph $H = (V, E_H)$, the ordinary graph $G = (V, E)$, where E is all the hyperedges in E_H composed of exactly two nodes, is defined to be the *underlying graph of* H. Throughout the remainder of this section, let $H = (V, E_H)$ be a hypergraph, $G = (V, E)$ be its underlying graph (E could be empty), and $b \geq 0$ be a nonnegative weighting of the nodes of G.

Let K be a clique in G, $b_K = \min_{v \in K} b_v$, and assume $b_K > 0$. A set of nodes $C \subseteq V \backslash K$ is a *stable cover of* K if C is a stable set in G and for each $u \in K$, there exists $h \in E_H$ such that $u \in h$ and $h \subseteq C \cup \{u\}$. C is a *minimal stable cover of* K if C is a stable cover of K such that $C \backslash \{v\}$ is not a stable cover of K for all $v \in C$ (examples of minimal stable covers are provided in Example 14). The *projection of* K is the hypergraph H_K obtained from H by adding all the hyperedges corresponding to minimal stable covers of K and letting b' be defined as

$$
b'_v = \begin{cases} b_v - b_K & \text{if } v \in K \\ b_v & \text{otherwise.} \end{cases}
$$

Theorem 13 *Let K be a clique in the underlying graph of the hypergraph $H = (V, E_H)$ such that $b_K > 0$. Let H_K be the clique projection of K.*

(1) Every feasible solution of BIP(H_K, b') is contained in a feasible solution of BIP(H_K, b') that includes a node in K.

(2) If S is a feasible solution of BIP(H, b) such that $S \cap K \neq \emptyset$, then S is a feasible solution of BIP(H_K, b').

(3) $\alpha_{b'}(H_K) \leq \alpha_b(H) - b_K$.

(4) *If there exists an optimal solution S of BIP(H, b) such that $S \cap K \neq \emptyset$, then $\alpha_{b'}(H_K) = \alpha_b(H) - b_K$.*

Proof.

(1) Let S be a feasible solution of BIP(H_K, b') such that $S \cap K = \emptyset$. For the sake of contradiction, suppose that $S \cup \{u\}$ is not a feasible solution of BIP(H_K, b') for each $u \in K$. Then every node in K is contained in a hyperedge whose hyperedge inequality is satisfied at equality by the incidence vector of S. Therefore S is a stable cover of K. Create a minimal stable cover C of K from S by removing nodes, one at a time, until C is minimal. But then S does not satisfy $\sum_{v \in C} x_v \leq |C| - 1$, which contradicts that S is a feasible solution of BIP(H_K, b') .

(2) Let S be a feasible solution of BIP(H, b) such that $S \cap K \neq \emptyset$. Let u be the unique node in $S \cap K$ and let C be a minimal stable cover of K in H. By definition, there exists $h \in E_H$ such that $u \in h$ and $h \subseteq C \cup \{u\}$. S must satisfy $\sum_{v \in h} x_v \leq |h| - 1$, therefore at least one node in h, say v, is not in S. But $v \in C$ because $h \subseteq C \cup \{u\}$ ($v \neq u$ because $u \in S$ but $v \notin S$). So S satisfies $\sum_{v \in C} x_v \leq |C| - 1$. Thus S satisfies all the constraints in BIP(H_K, b') .

(3) Let S be an optimal solution of BIP(H_K, b') . By (1), and the fact that $b' \geq 0$, we can assume that $S \cap K \neq \emptyset$. Let u be the unique node in $S \cap K$. Then

$$\begin{aligned}\alpha_{b'}(H_K) &= b'(S) = b'(S \backslash \{u\}) + b'_u \\ &= b(S \backslash \{u\}) + b_u - b_K = b(S) - b_K \\ &\leq \alpha_b(H) - b_K,\end{aligned}$$

where the final inequality follows from the fact that S also is a feasible solution of BIP(H, b) .

(4) From part (3), one needs only show that $\alpha_{b'}(H_K) \geq \alpha_b(H) - b_K$. Let S be an optimal solution of BIP(H, b) such that $S \cap K \neq \emptyset$ and let u be the unique node in $S \cap K$. Part (2) implies that S is a feasible solution of BIP(H_K, b') . Therefore

$$\begin{aligned}\alpha_{b'}(H_K) \geq b'(S) &= b'(S \backslash \{u\}) + b'_u \\ &= b(S \backslash \{u\}) + b_u - b_K \\ &= b(S) - b_K = \alpha_b(H) - b_K.\end{aligned}$$

■

To explain how Theorem 13 can be used to reduce a graph, suppose K is a clique in the underlying graph G and that there exists an optimal solution S of BIP(H, b) such that $S \cap K \neq \emptyset$ (refer to such a clique as *reducible).* This definition of reducible differs from the one given by Lovász and Plummer [23], but it generalizes the desired property of the clique from $\alpha(G') = \alpha(G) - 1$ to $\alpha_{b'}(H_K) = \alpha_b(H) - b_K$. Furthermore, any node v that has $b'_v = 0$ can be deleted from H_K because every inequality in BIP(H_K, b') has only nonnegative coefficients. Moreover, every hyperedge that contains v can be deleted from H_K, since the corresponding hyperedge inequality is redundant after x_v has been set to zero. The following example illustrates this approach.

Example 14 *Consider the graph depicted in Figure 7. It is straightforward to show that there exists an MSS that intersects $K_1 = \{1, 2, 3\}$. Thus K_1 can be deleted after it has been projected. Let H_1 be the projection of K_1. The projection creates the following hyperedge inequalities:*

$$\begin{aligned}x_4 + x_8 + x_{12} &\leq 2 \\ x_4 + x_9 + x_{11} &\leq 2 \\ x_5 + x_7 + x_{12} &\leq 2 \\ x_6 + x_7 + x_{11} &\leq 2 \\ x_5 + x_9 + x_{10} &\leq 2 \\ x_6 + x_8 + x_{10} &\leq 2.\end{aligned} \tag{1}$$

Figure 8 displays the underlying graph of H_1 after nodes 1, 2, and 3 have been deleted. It can now be shown that there exists an MSS in H_1 that intersects $K_2 = \{4, 5, 6\}$. Thus K_2 can be deleted by projecting it. Let H_2 be the projection of K_2. The minimal stable covers of K_2 are $\{8, 12\}, \{9, 11\}, \{7, 12\}, \{7, 11\}, \{9, 10\}$, and $\{8, 10\}$. When nodes 4, 5, and 6 are deleted, the hyperedge inequalities that were added when K_1 was projected are redundant and can be deleted. Thus H_2, shown in Figure 9 after nodes 4, 5, and 6 have been deleted, only contains ordinary edges, hence is an ordinary graph. In fact, H_2 is a clique, so no further reductions are necessary to solve the MSS problem.

For the graph in Figure 7, it also is possible to begin the reduction by projecting the clique $K'_1 = \{1, 4, 7, 10\}$. There are no stable covers of K'_1, so projecting K'_1 is the same as deleting it. Once K'_1 has been projected, then the clique $K'_2 = \{2, 5, 8, 11\}$ can be projected. There are no stable covers of K'_2, so projecting it is the same as deleting it. The remaining nodes, $\{3, 6, 9, 12\}$ form a clique, so no further reductions are necessary to solve the MSS problem.

When a reducible clique K is projected, it may be possible to eliminate nodes outside of K. If node $v \in V \backslash K$ is adjacent to every node in K in the underlying

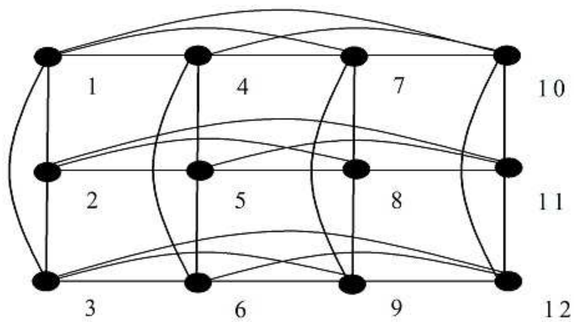

Fig. 7. Graph for Example 13.

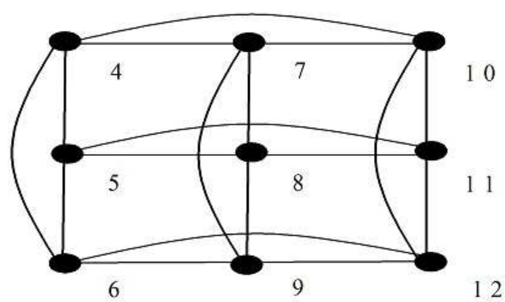

Fig. 8. The underlying graph of H_1 after $K = \{1, 2, 3\}$ has been projected and deleted.

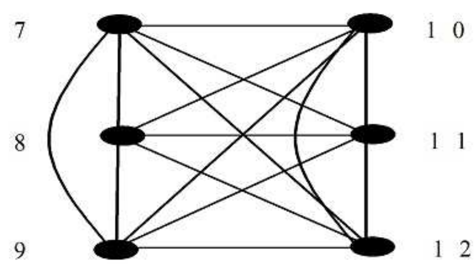

Fig. 9. H_2: The graph after $K_1 = \{1, 2, 3\}$ and $K_2 = \{4, 5, 6\}$ have been projected and deleted.

graph G, then v is a minimal stable cover of K. Hence the inequality $x_v \leq 0$ is placed in BIP(H_K, b') , which means that v can be eliminated. This corresponds to deleting N_{uv} when performing an edge projection on $e = (u, v)$. Consequently, if $K' \subseteq K$ is also reducible, then at least as many nodes will be eliminated by projecting K' as by projecting K.

Reducible cliques, as defined by Lovász and Plummer [23] and edge projections, as defined by Mannino and Sassano [25] and Mannino and Stefanutti [26] in the weighted case, are special cases of the clique projection defined here. Suppose K is a maximal clique in a graph $G = (V, E)$ and $\alpha(G[N(K)]) \leq 2$. K cannot have a minimal stable cover of size one because K is maximal, and it cannot have a minimal stable cover of size greater than two because $\alpha(G[N(K)]) \leq 2$. Hence projecting K will create an edge (if not already present) between every pair of nodes u and v such that $K \subseteq N(u) \cup N(v)$.

Furthermore, since every node has weight one, every node in K will have weight zero after the projection, hence can be deleted. Therefore, projecting K results in the same graph as defined by Lovász and Plummer. Now let $e = (u, v) \in E$ and define N_{uv}, N_u, and N_v the same as Mannino and Sassano. Projecting the clique $K = e$ will eliminate the nodes in N_{uv} because each of these nodes are a minimal stable cover of size one, as discussed above. Every other minimal stable cover is of the form $\{w, y\}$, where $w \in N_u$ and $y \in N_v$. Hence the projection will add an edge between every such pair of nodes. Furthermore, the weights of the nodes are modified in precisely the same manner. Consequently, projecting K results in the same graph as the edge projection of e, as defined in [25,26].

In general it is a difficult problem to determine if a clique is reducible. (If deciding reducibility of an arbitrary clique can be done in polynomial time, then deciding whether or not a node u is contained in an MSS can be done in polynomial time, which would yield a polynomial time algorithm for the MSS problem.) The following theorem gives two different sets of sufficient conditions that can be used to determine if a given clique is reducible.

Theorem 15 *Let K be a clique in the underlying graph $G = (V, E)$ of hypergraph $H = (V, E_H)$.*

(1) If K does not have any stable covers in H, then K is reducible.

(2) Suppose $u \in K$, $N'(H, u) = \{v \in V \setminus K : \exists h \in E_H \ni \{u, v\} \subseteq h\}$, and H_u is the hypergraph induced by $N'(H, u)$. If $b_u \geq \alpha_b(H_u)$, then K is reducible.

Proof.

(1) Since K does not have any stable covers, the only difference between BIP(H, b) and BIP(H_K, b') is the objective function, i.e., no constraints have been added to BIP(H, b) to obtain BIP(H_K, b'). Therefore, if S is an optimal solution of BIP(H, b), then S is a feasible solution of BIP(H_K, b'). Thus, Theorem 13(1) implies that S is contained in a feasible solution S' of BIP(H_K, b') that includes a node in K. Clearly, S' also is a feasible solution of BIP(H_K, b) and $b(S') \geq b(S)$. Hence every optimal solution of BIP(H, b) is contained in an optimal solution that intersects K. Therefore, K is reducible.

(2) Let S be an optimal solution of BIP(H, b) such that $S \cap K = \emptyset$. The set $S' = (S \backslash N'(H, u)) \cup \{u\}$ is also a feasible solution of BIP(H, b). Furthermore,

$$\begin{aligned} b(S') &= b((S \backslash N'(H, u)) \cup \{u\}) \\ &= b(S \backslash N'(H, u)) + b_u \\ &= b(S) - b(S \cap N'(H, u)) + b_u \\ &\geq b(S) - \alpha_b(H_u) + b_u \geq b(S) = \alpha_b(H), \end{aligned}$$

so S' is an optimal solution of BIP(H, b) that intersects K. Therefore, K is reducible.

■

The graph in Figure 8 has a clique $K = \{1, 4, 7, 10\}$ that does not have any stable covers. So Theorem 15(1) implies that K is reducible. All three triangles in the graph in Figure 5 satisfy the conditions of Theorem 15(2), so they are all reducible.

A *Steiner Triple System* consists of a set $D = \{1, 2, ..., n\}$ and a collection of triples, which are subsets of D of size three, such that every pair of elements in D is contained in exactly one triple. Fulkerson et al. [17] created two computationally difficult set covering problems arising from Steiner Triple Systems. Subsequently, these problems were converted into equivalent MSS problems, and several more problems of the same type were generated. Four such problems were included in the benchmark graphs for the Second DIMACS Implementation Challenge on Clique, Graph Coloring, and Satisfiability. These problems have indeed proven to be difficult. To date, no exact stable set algorithm has been able to directly solve the largest such problem, mann_81, although Mannino and Sassano [25] were able to solve it indirectly.

The power of clique projections is illustrated on these challenging problems. As shown in Table 5, the clique projections produce a large reduction in both the number of nodes (variables) and edges (constraints). These reductions were obtained by using Theorem 15(2) to project all the triangles, which correspond to the Steiner triples, in the graph.

Acknowledgments

This research has been supported in part by the Air Force Office of Scientific Research (FA9550-07-1-0232, FA9550-10-1-0387). The authors wish to thank the Editor and the anonymous referee for their helpful comments and suggestions, which has resulted in a significantly improved manuscript. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

Table 5

	Number of Nodes		Number of Edges	
Graph	**Originally**	**After Projections**	**Originally**	**After Projections**
mann_a9	45	9	72	12
mann_a27	378	27	702	117
mann_a45	1,035	45	1,980	330
mann_a81	3,321	81	6,480	1,080

Clique Projections Applied to MSS Problems Arising from Steiner Triple Systems.

References

[1] G. Alexe, P.L. Hammer, V.V. Lozin, and D. de Werra. Struction revisited. *Discrete Applied Mathematics*, 132:27–46, 2004.

[2] L. Babel. Finding maximum cliques in arbitrary and in special graphs. *Computing*, 46:321–341, 1991.

[3] L. Babel. A fast algorithm for the maximum weight clique problem. *Computing*, 52:31–38, 1994.

[4] L. Babel and G. Tinhofer. A branch and bound algorithm for the maximum clique problem. *ZOR-Methods and Models of Operations Research*, 34:207–217, 1990.

[5] E. Balas and J. Xue. Minimum weighted coloring of triangulated graphs, with application to maximum weight vertex packing and clique finding in arbitrary graphs. *SIAM Journal on Computing*, 20(2):209–221, 1991.

[6] E. Balas and C. S. Yu. Finding a maximum clique in an arbitrary graph. *SIAM Journal on Computing*, 15(4):1054–1068, 1986.

[7] R. Beigel. Finding maximum independent sets in sparse and general graphs. In *Proceedings of the tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages S856–S857. Society for Industrial and Applied Mathematics, 1999.

[8] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization*, pages 1–74. Kluwer Academic Publishers, Dordrecht, 1999.

[9] E. Boros, P. L. Hammer, and G. Tavares. Preprocessing of Unconstrained Quadratic Binary Optimization. Technical report RRR 10-2006, RUTCOR, 2006.

[10] J. M. Bourjolly, P. Gill, G. Laporte, and H. Mercure. An exact quadratic 0–1 algorithm for the stable set problem. In D. S. Johnson and M. A. Trick, editors, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Volume 26: Cliques, Coloring, and Satisfiability*, pages 53–73. American Mathematical Society, Providence, RI, 1996.

[11] J. M. Bourjolly and W. R. Pulleyblank. König-Egerváry graphs, 2-bicritical graphs and fractional matchings. *Discrete Applied Mathematics*, 24:63–82, 1989.

[12] R. Carraghan and P. M. Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9:375–382, 1990.

[13] J. Chen, I. Kanj, and W. Jia. Vertex cover: Further observations and further improvements. In *Lecture Notes in Computer Science 1665 (WG'99)*, pages 313–324, Berlin, 1999. Springer.

[14] J. Chen, L. Liu, and W. Jia. Improvement on vertex cover for low-degree graphs. *Networks*, 35(4):253–259, 2000.

[15] Ch. Ebenegger, P.L. Hammer, and D. de Werra. Pseudo-boolean functions and stability of graphs. *Annals of Discrete Mathematics*, 19:83–98, 1984.

[16] C. Friden, A. Hertz, and D. de Werra. TABARIS: An exact algorithm based on tabu search for finding a maximum independent set in a graph. *Computers and Operations Research*, 17(5):437–445, 1990.

[17] D. R. Fulkerson, G. L. Nemhauser, and L.E. Trotter, Jr. Two computationally difficult set covering problems that arise in computing the 1-width of incidence matrices of Steiner triple systems. *Mathematical Programming Study*, 2:72–81, 1974.

[18] P.L. Hammer, P. Hansen, and B. Simeone. Vertices belonging to all or to no maximum stable sets of a graph. *SIAM Journal on Algebraic and Discrete Methods*, 3(4):511–522, 1982.

[19] T. Jian. An $o(2^{0.304n})$ algorithm for solving maximum independent set problems. *IEEE Trans. Comput.*, C-35:847–851, 1986.

[20] E. L. Johnson and M. W. Padberg. Degree-two inequalities, clique facets, and biperfect graphs. *Annals of Discrete Mathematics*, 16:169–187, 1982.

[21] L. Lipták and L. Lovász. Facets with fixed defect of the stable set polytope. *Mathematical Programming, Series A*, 88:33–44, 2000.

[22] L. Lipták and L. Lovász. Critical facets of the stable set polytope. *Combinatorica*, 21:61–88, 2001.

[23] L. Lovász and M.D. Plummer. *Matching Theory*. Akademiai Kiado, Budapest and North-Holland, Amsterdam, 1986.

[24] C. Mannino and A. Sassano. An exact algorithm for the maximum stable set problem. *Journal of Combinatorial Optimization and Applications*, 3:243–258, 1994.

[25] C. Mannino and A. Sassano. Edge projection and the maximum cardinality stable set problem. In D. S. Johnson and M. A. Trick, editors, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Volume 26: Cliques, Coloring, and*

Satisfiability, pages 205–219. American Mathematical Society, Providence, RI, 1996.

[26] C. Mannino and E. Stefanutti. An augmentation algorithm for the maximum weighted stable set problem. *Computational Optimization and Applications*, 14:367–381, 1999.

[27] G.L. Nemhauser and G. Sigismondi. A strong cutting plane/branch-and-bound algorithm for node packing. *Journal of the Operational Research Society*, 43(5):443–457, 1992.

[28] G.L. Nemhauser and L.E. Trotter, Jr. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.

[29] P. R. J. Ostergård. A new algorithm for the maximum-weight clique problem. *Nordic Journal of Computing*, 8:424–436, 2001.

[30] P. R. J. Ostergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120:197–207, 2002.

[31] P. M. Pardalos and N. Desai. An algorithm for finding a maximum weighted independent set in an arbitrary graph. *Intern. J. Computer Math.*, 38:163–175, 1991.

[32] P. M. Pardalos and G. P. Rodgers. A branch and bound algorithm for the maximum clique problem. *Computers and Operations Research*, 19:363–375, 1992.

[33] J. C. Régin. Using constraint programming to solve the maximum clique problem. In *Principles and Practice of Constraint Programming — CP 2003*, pages 634–648, Heidelberg, 2003. Lecture Notes in Computer Science, Springer-Verlag.

[34] J. M. Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7:425–440, 1986.

Received 22-12-2009; revised 27-10-2010; accepted 21-11-2010

[35] J. M. Robson. Finding a maximum independent set in time $o(2^{n/4})$. Technical report, 2001.

[36] A. Sassano. Reducible cliques and the strong perfect graph conjecture. Technical Report IASI-Report R. 257, 1989.

[37] A. Schrijver. *Combinatorial Optimization*. Springer-Verlag, 2003.

[38] E. C. Sewell. *Stability Critical Graphs and the Stable Set Polytope*. PhD thesis, Cornell University, School of ORIE, Ithaca, New York 14850, 1990.

[39] E. C. Sewell. Binary integer programs with two variables per inequality. *Mathematical Programming*, 75:467–476, 1996.

[40] E. C. Sewell. A branch and bound algorithm for the stability number of a sparse graph. *INFORMS Journal on Computing*, 10:438–447, 1998.

[41] E. C. Sewell and L.E. Trotter, Jr. Stability critical graphs and even subdivisions of K_4. *Journal of Combinatorial Theory, Series B*, 59(1):74–84, 1993.

[42] M. Shindo and E. Tomita. A simple algorithm for finding a maximum clique and its worst-case time complexity. *Systems and Computers in Japan*, 21(3):1–13, 1990.

[43] C. De Simone and A. Sassano. Stability number of bull- and chair-free graphs. *Discrete Applied Mathematics*, 41:121–129, 1993.

[44] U. Stege and M. Fellows. An improved fixed-parameter-tractable algorithm for vertex cover. Department of Computer Science Tech. Report 318, ETH Zurich, 1999.

[45] R. E. Tarjan and A. E. Trojanowski. Finding a maximum independent set. *SIAM Journal on Computing*, 6:537–546, 1977.

[46] D. R. Wood. An algorithm for finding a maximum clique in a graph. *Operations Research Letters*, 21:211–217, 1997.