Clique-based facets for the precedence constrained knapsack problem

Natashia Boland * Andreas Bley † Christopher Fricke ‡
Gary Froyland § Renata Sotirov ¶

Abstract

We consider a knapsack problem with precedence constraints imposed on pairs of items, known as the precedence constrained knapsack problem (PCKP). This problem has applications in manufacturing and mining, and also appears as a subproblem in decomposition techniques for network design and related problems. We present a new approach for determining facets of the PCKP polyhedron based on clique inequalities. A comparison with existing techniques, that lift knapsack cover inequalities for the PCKP, is also presented. It is shown that the clique-based approach generates facets that cannot be found through the existing cover-based approaches, and that the addition of clique-based inequalities for the PCKP can be computationally beneficial.

1 Introduction

In this paper, we consider the polyhedral structure of the *precedence constrained knapsack problem* (PCKP), also known as the *partially ordered knapsack problem*. Let $\mathcal{N}$ be a set of items and $\mathcal{S} \subseteq \mathcal{N} \times \mathcal{N}$ denote a partial order or set of precedence relationships on the items. A precedence relationship $(i, j) \in \mathcal{S}$ exists if item i can be placed in the knapsack only if item j is in the knapsack. Each item $i \in \mathcal{N}$ has a value $c_i \in \mathbb{Z}$ and a weight $a_i \in \mathbb{Z}^+$, and the knapsack has a capacity $b \in \mathbb{Z}^+$. The PCKP is the problem of finding a maximum value subset of $\mathcal{N}$ whose total weight does not exceed the knapsack capacity, and that also satisfies the precedence relationships.

*School of Mathematical and Physical Sciences, University of Newcastle, Australia (Natashia.Boland@newcastle.edu.au)

†Technical University Berlin, Straße des 17. Juni 136, D-10623 Berlin, Germany (bley@math.tu-berlin.de)

‡TSG Consulting, Level 11, 350 Collins Street, Melbourne VIC 3000, Australia

§School of Mathematics and Statistics, University of New South Wales, Australia (g.froyland@unsw.edu.au)

¶Universiteit van Tilburg Warandelaan 2, P.O. Box 90153, 5000 LE Tilburg, The Netherlands (r.sotirov@uvt.nl)

The precedence constraints can be represented by the directed graph $G = (\mathcal{N}, \mathcal{S})$, where the node set is the set of all items $\mathcal{N}$, and each precedence constraint in $\mathcal{S}$ is represented by a directed arc. If G contains a cycle, all nodes within the cycle must either all be included in, or all be excluded from, the knapsack. Hence the cycle can be contracted into a single node, with cumulative value and weight coefficients, and the resulting directed graph is acyclic. Thus without loss of generality we assume that G is acyclic. Note that the precedence constraints are transitive, so without loss of generality we assume that $\mathcal{S}$ does not contain any redundant relationships, that is, $\mathcal{S}$ is the set of all immediate predecessor arcs.

An integer programming formulation of the PCKP is as follows. Let

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is included in the knapsack} \\ 0, & \text{otherwise} \end{cases} \qquad \text{for all } i \in \mathcal{N}.$$

Then the PCKP may be written as:

$$\text{(PCKP)} \qquad \begin{aligned} \max \sum_{i \in \mathcal{N}} c_i x_i & & & \quad (1) \\ \text{s.t.} \sum_{i \in \mathcal{N}} a_i x_i &\leq b & & \quad (2) \\ x_i &\leq x_j & \text{for all } (i,j) \in \mathcal{S} & \quad (3) \\ x_i &\in \{0,1\} & \text{for all } i \in \mathcal{N}. & \quad (4) \end{aligned}$$

The PCKP appears in a wide range of applications. These include investment problems (Ibarra and Kim [10]), production planning (Stecke and Kim [15]), strip mining (Johnson and Niemi [11]) and local access telecommunication network design (Shaw *et al.* [14]). In all of these cases the underlying precedence graph has a special structure such as a tree and the PCKP has been solved using dynamic programming algorithms or heuristics.

Garey and Johnson [9] showed that the PCKP is NP-complete. The polyhedral structure of the problem was first investigated by Boyd [6], who extended the concept of a cover inequality for the standard 0-1 knapsack polyhedron to the PCKP polyhedron. Further investigation of the PCKP polyhedron is presented by both Park and Park [13] and van de Leensel *et al.* [16], where lifting orders and general sequential lifting procedures are derived to lift valid knapsack cover-based inequalities from lower dimensional polyhedrons into facets of the PCKP polyhedron.

In this paper, we determine facet-defining inequalities for the polyhedron defined by the feasible solutions of (2)-(4). Unlike previous work [6, 13, 16], we do not take knapsack covers as our starting point, but instead investigate clique inequalities derived from a graph representing pairwise conflict relationships between variables.

We begin in Section 2 by introducing the notation and definitions used throughout the paper. We also derive properties of the precedence relationships that will be useful in our investigation, and present the concept of a conflict graph. In Section 3 we introduce clique inequalities for the PCKP, and derive necessary and sufficient conditions under which they represent facets of $conv(P)$. A comparison of clique inequalities and the results of Boyd [6], Park and Park [13], and van de Leensel *et al.* [16], which, as already mentioned, are all based on knapsack cover-like inequalities, is presented in Section 4. We provide a more complete classification of these knapsack cover-like inequalities than has previously been given. The differences, similarities, and computational strength of the various classes of constraints are illustrated in examples in Section 5. We demonstrate that our clique-based approach can generate facet-defining inequalities for $conv(P)$, without the need for the computationally expensive lifting procedures that are used in existing cover-based approaches. In Section 6 we apply clique inequalities in both a cutting plane approach at the root node of the branch and bound tree and in a branch and cut framework for large realistic PCK instances, and demonstrate significant computational benefits.

2 Notation and Properties of the Precedence Relationships

A summary of the notation used throughout this paper is given in Table 9 at the end of the paper. For each $(i, j) \in \mathcal{S}$, item i is an *immediate predecessor* of item j and item j is an *immediate successor* of item i. Let S_i be the set of immediate predecessors of item i, that is let $S_i = \{j \in \mathcal{N} : (i, j) \in \mathcal{S}\}$. It follows that the set of all precedence relationships $\mathcal{A}$ is the transitive closure of $\mathcal{S}$, and $(i, j) \in \mathcal{A}$ if and only if there exists a path from node i to node j in the directed acyclic graph $G = (\mathcal{N}, \mathcal{S})$. Let A_i be the *minimal set of items*, including item i, that must be included in the knapsack for item i to be included, that is $A_i = \{j \in \mathcal{N} : (i, j) \in \mathcal{A}\} \cup \{i\}$. Note that inclusion in the set A_i is also transitive, so if $j \in A_k$ and $k \in A_i$ then $j \in A_i$. Property 1 follows directly.

Property 1 *Let $i \in \mathcal{N}$. For all $j \in A_i$ it must be that $A_j \subseteq A_i$.*

Consider a set of items $B \subseteq \mathcal{N}$. Let $A(B) = \cup_{i \in B} A_i$. Then $A(B)$ gives the minimal set of items that must be included for all items in the set B to be included in the knapsack. Now consider the set of items that cannot be included unless item i has been included in the knapsack, and include item i in this set. This is the set of all successors of item i, which we denote as D_i, hence $D_i = \{j \in \mathcal{N} : i \in A_j\}$. By the transitivity of inclusion in the A_i sets, it follows directly that for any item $j \in A_i$, it must be that $i \in D_j$. Hence, given a set of items $\mathcal{N}$ and the immediate predecessor set S_i for each $i \in \mathcal{N}$, the corresponding entire precedence sets A_i and entire successor sets D_i can be deduced for each $i \in \mathcal{N}$. Note that item i is included in both the entire precedence set A_i and the entire successor set D_i.

For a given set $B \subseteq \mathcal{N}$ we also require the concept of a subset of B that contains successors of an item $k \in A(B) \setminus B$.

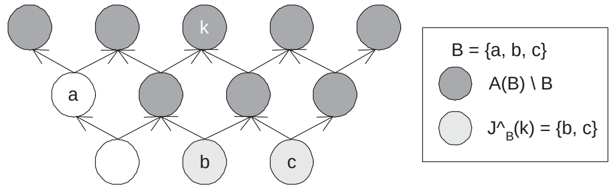


Figure 1: Illustration of a descendent set $\hat{J}_B(k)$.

Definition 1 *For each set $B \subseteq \mathcal{N}$ with $A(B) \setminus B \neq \emptyset$, for each $k \in A(B) \setminus B$ there exists $j \in B$ such that $k \in A_j$. Let $\hat{J}_B(k)$ denote these j, that is $\hat{J}_B(k) = \{j \in B : k \in A_j\} = B \cap D_k$ for each $k \in A(B) \setminus B$. We refer to $\hat{J}_B(k)$ as the* ***descendent set of** k **in** B.*

See Figure 1 for an illustration of a descendent set $\hat{J}_B(k)$. In all diagrams throughout this paper, we show the set of immediate predecessors S_i for all $i \in \mathcal{N}$. The A_i sets can be deduced by finding the transitive closure of the S_i sets.

We now combine the precedence sets with the knapsack constraint (2) to determine the minimum capacity required to include each item in the knapsack. Let $H(B) = \sum_{j \in A(B)} a_j$ be the total capacity required to include the items in the set B in the knapsack. It follows that $H(\{i\}) = \sum_{j \in A_i} a_j$ is the capacity required to include item i in the knapsack. For ease of notation let $H_i = H(\{i\})$. We assume that for every item, there exists a feasible solution in which it is included in the knapsack. Otherwise, the item can be deleted from the problem instance.

Assumption 1 *Each item in the set $\mathcal{N}$ could be included in the knapsack, that is $H_i \leq b$ for all $i \in \mathcal{N}$.*

It follows directly from Assumption 1 that the PCKP polyhedron is full-dimensional. We now determine when the inclusion of a given set of items $B \subseteq \mathcal{N}$ in the knapsack is feasible. In what follows, e_i is the i^{th} standard basis vector in $\mathbb{R}^{|\mathcal{N}|}$. For any set $B \subseteq \mathcal{N}$, let $x(B) \in \{0,1\}^{|\mathcal{N}|}$ denote the characteristic vector of B, that is $x(B) = \sum_{i \in B} e_i$.

Definition 2 *We say that the set of items $B \subseteq \mathcal{N}$ is a* ***feasible packing*** *of the knapsack if*

(i) for all $j \in B$, $A_j \subseteq B$, and

(ii) $H(B) = \sum_{j \in A(B)} a_j \leq b$.

We now provide a series of technical results regarding feasible packings and precedence sets. These help to simplify the proofs of our main results in Section 3. We omit proofs, other than that of the last result, which appears in an appendix, as they are all straightfoward consequences of the above definitions.

Lemma 1

(i) *Let $B \subseteq \mathcal{N}$. If $i \in A(B)$ then $A_i \subseteq A(B)$.*

(ii) *Let $j \in \mathcal{N} \setminus D_i$ for some $i \in \mathcal{N}$. Then $A_j \subseteq \mathcal{N} \setminus D_i$.*

(iii) *If $B \subseteq \mathcal{N}$ with $H(B) \leq b$, then $A(B)$ is a feasible packing.*

(iv) *Let $B_1, \ldots, B_m,\ m \in \mathbb{Z}^+$ be a collection of feasible packings such that $H(B_1 \cup \ldots \cup B_m) \leq b$. Then $B_1 \cup \ldots \cup B_m$ is a feasible packing.*

(v) *Let $B \subseteq \mathcal{N}$ with $H(B) \leq b$. Then $A(B) \setminus \{i\}$ is a feasible packing for any $i \in B$ satisfying $A(B \setminus \{i\}) \subsetneq A(B)$.*

If $B \subseteq \mathcal{N}$ with $H(B) \leq b$ then we say that B **induces** the feasible packing $A(B)$. Note that we often apply Lemma 1(iii) and (v) with $B = \{k\}$ to assert that A_k and $A_k \setminus \{k\}$ are feasible packings, respectively.

2.1 Conflict Graphs and their Properties

In order to identify potential facet-defining inequalities for $conv(P)$, we require the following definition of a conflict graph for the instance of the PCKP under consideration.

Definition 3 *A* ***conflict graph*** *$CG = (\mathcal{N}, E)$ contains the edge $\{i, j\} \in E$ if and only if the pair of items $i, j \in \mathcal{N}$* ***cannot*** *be included in the knapsack together, that is if and only if $H(\{i, j\}) > b$.*

In all illustrations of a conflict graph throughout this paper, we show only nodes that are not singletons. A **clique** $\mathcal{C} \subseteq \mathcal{N}$ in the conflict graph CG is a set of nodes such that every pair of nodes in $\mathcal{C}$ is joined by an edge. Hence each pair of items in $\mathcal{C}$ cannot be included in the knapsack simultaneously, and it follows that at most one item in $\mathcal{C}$ can be included in the knapsack. A **maximal clique** is a clique that cannot be enlarged by adding any additional node. We now derive technical properties of cliques in the conflict graph, useful in our main results in Section 3. Proofs are given in an appendix.

Lemma 2 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a clique in the conflict graph CG.*

(i) *For each $i \in \mathcal{C}$, $A_i \cap \mathcal{C} = \{i\}$.*

(ii) *For each $i \in \mathcal{C}$, $A_i \setminus \{i\} \subseteq A(\mathcal{C}) \setminus \mathcal{C}$.*

(iii) *For each $k \in A(\mathcal{C}) \setminus \mathcal{C}$, $A_k \subseteq A(\mathcal{C}) \setminus \mathcal{C}$.*

The following lemma, using results from both Lemmas 1 and 2 and proved in an appendix, simplifies the proofs of our two main results.

Lemma 3 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a clique in the conflict graph CG. Let $h \in \mathcal{C}$ and $k \notin \mathcal{C} \cup A_h$ be such that $H(\{h, k\}) \leq b$. Then* *(i) both* $A_h \cup A_k$ *and* $(A_h \cup A_k) \setminus \{k\}$ *are feasible packings; and*

(ii) $(A_h \cup A_k) \cap \mathcal{C} = ((A_h \cup A_k) \setminus \{k\}) \cap \mathcal{C} = \{h\}$.

Along with these properties of the PCKP, we also require general results from polyhedral theory. In particular, we require the following lemma, which is straightforward to prove.

Lemma 4 *Let* F *and* $\bar{F}$ *be two faces of a non-empty polyhedron* Q*, and let* $F \subsetneqq \bar{F} \subsetneqq Q$*. Then* F *cannot represent a facet of* Q*.*

3 Clique-Based Facets for the PCKP Polyhedron

The properties discussed in Section 2 are now used to identify facets of $conv(P)$, where $CG = (\mathcal{N}, E)$ is a conflict graph determined according to Definition 3. The following result is obvious.

Lemma 5 *Let* $\mathcal{C} \subseteq \mathcal{N}$ *be a clique in the conflict graph* CG*. Then the clique inequality*

$$\sum_{j \in \mathcal{C}} x_j \leq 1 \tag{5}$$

is valid for P*.*

Definition 4 *Let* $\mathcal{C} \subseteq \mathcal{N}$ *be a clique in the conflict graph* CG*. Let* $\mathcal{P}(\mathcal{C})$ *be the set of items in the intersection of the entire precedence sets of all the items in the clique* $\mathcal{C}$*, that is* $\mathcal{P}(\mathcal{C}) = \bigcap_{j \in \mathcal{C}} A_j$*.*

In the case where $\mathcal{P}(\mathcal{C}) \neq \emptyset$, we will see that we can replace the right-hand side of (5) with an element of $\mathcal{P}(\mathcal{C})$. We consider the cases of $\mathcal{P}(\mathcal{C}) = \emptyset$ and $\mathcal{P}(\mathcal{C}) \neq \emptyset$ separately.

3.1 Case 1: Empty intersection set, $\mathcal{P}(\mathcal{C}) = \emptyset$

In this case, we are able to determine necessary and sufficient conditions under which (5) is facet-defining for $conv(P)$. We also give a straightforward procedure that, given any maximal clique $\mathcal{C} \subseteq \mathcal{N}$ with $\mathcal{P}(\mathcal{C}) = \emptyset$, can generate a maximal clique satisfying these conditions.

Definition 5 *Let* $\mathcal{C} \subseteq \mathcal{N}$ *be a clique in the conflict graph* CG*. Let* $F_{\mathcal{C}} = \left\{ x \in conv(P) : \sum_{j \in \mathcal{C}} x_j = 1 \right\}$*, that is,* $F_{\mathcal{C}}$ *represents the face of* $conv(P)$ *determined from the valid clique inequality (5).*

The necessary and sufficient conditions on $\mathcal{C}$ so that $F_{\mathcal{C}}$ is facet-defining for $conv(P)$ are given by Condition 1.

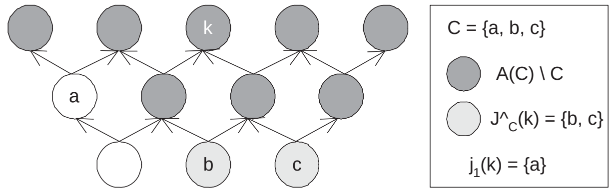


Figure 2: Illustration of the definition of $j_1(k)$ from Condition 1 for Case 1, $\mathcal{P}(\mathcal{C}) = \emptyset$, on the precedence graph $(\mathcal{N}, \mathcal{S})$. Take $a_i = 1$ for all $i \in \mathcal{N}$ and $b = 7$.

Condition 1 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a maximal clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) = \emptyset$. Either $A(\mathcal{C}) = \mathcal{C}$, or for every $k \in A(\mathcal{C}) \setminus \mathcal{C}$ there exists $j_1(k) \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)$ such that $H(\{j_1(k), k\}) \leq b$.*

See Figure 2 for an illustration of $j_1(k)$ when $A(\mathcal{C}) \setminus \mathcal{C} \neq \emptyset$ in Case 1.

Suppose a maximal clique $\mathcal{C} \subseteq \mathcal{N}$ in the conflict graph CG with $\mathcal{P}(\mathcal{C}) = \emptyset$ is given, and Condition 1 does not hold. The following lemma shows that in this case (5) is redundant in the description of $conv(P)$. It also provides a way to construct another maximal clique $\mathcal{C}'$ from $\mathcal{C}$ for which Condition 1 holds.

Lemma 6 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a maximal clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) = \emptyset$. Suppose Condition 1 does not hold; that is, $A(\mathcal{C}) \neq \mathcal{C}$ and for some $k \in A(\mathcal{C}) \setminus \mathcal{C}$, $H(\{j, k\}) > b$ for all $j \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)$. Then*

(i) $\mathcal{C}' = (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\}$ is also a maximal clique in CG, $\mathcal{P}(\mathcal{C}') = \emptyset$, and $F_{\mathcal{C}} \nsubseteq F_{\mathcal{C}'}$; and

(ii) The clique inequality (5) for $\mathcal{C}$ is redundant in the description of $conv(P)$.

Proof. We first prove part (i). Since $\mathcal{P}(\mathcal{C}) = \emptyset$, we have that $\hat{J}_{\mathcal{C}}(k) \subsetneq \mathcal{C}$. Let $\mathcal{C}' = (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\}$. Then $|\mathcal{C}'| \geq 2$ and $\mathcal{C}'$ is also a clique in $CG = (\mathcal{N}, E)$. Suppose that $\mathcal{C}'$ is not a maximal clique in CG, so there exists $i \notin \mathcal{C}'$ such that $\{i, j\} \in E$ for all $j \in \mathcal{C}'$. Note that $i \notin \hat{J}_{\mathcal{C}}(k)$ since otherwise $k \in A_i$ and $\{i, k\} \notin E$ by Assumption 1. Since $\{i, k\} \in E$ and by Definition 1 we have that $A_k \subseteq A_h$ for all $h \in \hat{J}_{\mathcal{C}}(k)$, it follows that $\{i, h\} \in E$ for all $h \in \hat{J}_{\mathcal{C}}(k)$. So we have $\{i, j\} \in E$ for all $j \in \mathcal{C}' \cup \hat{J}_{\mathcal{C}}(k) = (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\} \cup \hat{J}_{\mathcal{C}}(k)$. In particular, $\{i, j\} \in E$ for all $j \in \mathcal{C}$, which contradicts the maximality of $\mathcal{C}$. Hence $\mathcal{C}' = (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\}$ is also a maximal clique in CG.

By definition we have that $k \in A_j$ for all $j \in \hat{J}_{\mathcal{C}}(k)$, and hence by Lemma 1(i) $A_k \subseteq A_j$ for all $j \in \hat{J}_{\mathcal{C}}(k)$. It follows that $A_k \subseteq \cap_{j \in \hat{J}_{\mathcal{C}}(k)} A_j$. Hence $\mathcal{P}(\mathcal{C}') = \cap_{j \in (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\}} A_j = (\cap_{j \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)} A_j) \cap A_k \subseteq (\cap_{j \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)} A_j) \cap (\cap_{j \in \hat{J}_{\mathcal{C}}(k)} A_j) = \cap_{j \in \mathcal{C}} A_j = \mathcal{P}(\mathcal{C}) = \emptyset$.

Let $G_{\mathcal{C}} = \{x \in \mathbb{R}^{|\mathcal{N}|} : \sum_{j \in \mathcal{C}} x_j = 1\}$ and $G_{\mathcal{C}'} = \{x \in \mathbb{R}^{|\mathcal{N}|} : \sum_{j \in \mathcal{C}'} x_j = 1\}$. Let $I_{\mathcal{C}} = P \cap G_{\mathcal{C}}$ and $I_{\mathcal{C}'} = P \cap G_{\mathcal{C}'}$. We will show that $I_{\mathcal{C}} \subsetneq I_{\mathcal{C}'}$. Let $x \in I_{\mathcal{C}}$. Then there is exactly one $j \in \mathcal{C}$ such that $x_j = 1$. If $j \in \hat{J}_{\mathcal{C}}(k)$ then $k \in A_j$ and $x_k = 1$, implying that $x \in I_{\mathcal{C}'}$. If $j \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)$ then obviously $x \in I_{\mathcal{C}'}$. Hence

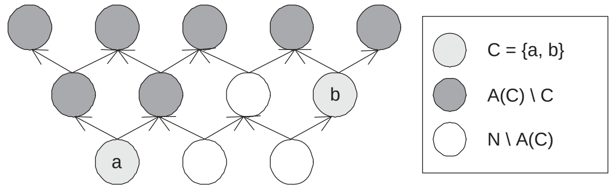


Figure 3: Illustration of the situations for consideration in Case 1, $\mathcal{P}(\mathcal{C}) = \emptyset$.

$I_{\mathcal{C}} \subseteq I_{\mathcal{C}'}$. Now consider the feasible packing A_k. Then $x(A_k) \in I_{\mathcal{C}'}$. However, since $k \in A(\mathcal{C}) \setminus \mathcal{C}$, by Lemma 2(iii) we have that $A_k \subseteq A(\mathcal{C}) \setminus \mathcal{C}$, and hence $A_k \cap \mathcal{C} = \emptyset$. Thus $x(A_k) \notin I_{\mathcal{C}}$. Hence $I_{\mathcal{C}} \subsetneqq I_{\mathcal{C}'}$. Since $I_{\mathcal{C}}$ and $I_{\mathcal{C}'}$ are sets of binary vectors, it follows that $conv(I_{\mathcal{C}}) \subsetneqq conv(I_{\mathcal{C}'})$.

Now observe that $F_{\mathcal{C}} = conv(P) \cap G_{\mathcal{C}}$ and $F_{\mathcal{C}'} = conv(P) \cap G_{\mathcal{C}'}$, and furthermore, by Lemma 5, the hyperplanes $G_{\mathcal{C}}$ and $G_{\mathcal{C}'}$ are defined by *valid* inequalities for P. Thus by Lemma 6.1.1 of Balas [2] we have that $conv(I_{\mathcal{C}}) = conv(P \cap G_{\mathcal{C}}) = conv(P) \cap G_{\mathcal{C}} = F_{\mathcal{C}}$, and that $conv(I_{\mathcal{C}'}) = conv(P \cap G_{\mathcal{C}'}) = conv(P) \cap G_{\mathcal{C}'} = F_{\mathcal{C}'}$, and hence $F_{\mathcal{C}} \subsetneqq F_{\mathcal{C}'}$.

We now prove part (ii). Note that $0 \in P$ but $0 \notin F_{\mathcal{C}'}$, hence $F_{\mathcal{C}'} \subsetneqq conv(P)$. Part (ii) now follows since by Lemma 4 one has that $F_{\mathcal{C}}$ cannot represent a facet of $conv(P)$, and hence the clique inequality (5) for $\mathcal{C}$ is redundant in the description of $conv(P)$. ■

As a consequence of Lemma 6, Condition 1 is a necessary condition for the inequality (5) to be facet defining. The next theorem states that this condition is also sufficient.

Theorem 1 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a maximal clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) = \emptyset$. Then $F_{\mathcal{C}}$ from Definition 5 is a facet of $conv(P)$ if and only if Condition 1 holds.*

Proof.

($\Rightarrow$) This is a direct consequence of Lemma 6.

($\Leftarrow$) Consider $\mathcal{C} \subseteq \mathcal{N}$ such that $\mathcal{C}$ is a maximal clique in the conflict graph CG, $\mathcal{P}(\mathcal{C}) = \emptyset$, and Condition 1 holds. Suppose $\lambda\, x = \lambda_0$ for all $x \in F_{\mathcal{C}}$ holds for some (λ, λ_0), where $F_{\mathcal{C}}$ is given by Definition 5. If it can be shown that $\lambda_k = \lambda_0$ for all $k \in \mathcal{C}$, and $\lambda_k = 0$ otherwise, then by Theorem 3.6 of Nemhauser and Wolsey [12] we will have proved that $F_{\mathcal{C}}$ represents a facet of $conv(P)$. There are three cases to consider, described in detail below and illustrated in Figure 3.

1. **Case 1(a)**: Let $k \notin \mathcal{C}$.

   There are two subcases: $k \notin A(\mathcal{C})$ and $k \in A(\mathcal{C}) \setminus \mathcal{C}$. In the first case, since $\mathcal{C}$ is maximal there must exist at least one $h \in \mathcal{C}$ such that $H(\{h, k\}) \le b$. Note that since $k \notin A(\mathcal{C})$ and $h \in \mathcal{C}$ then $k \notin A_h$. In the second case, by Condition 1 there exists $j_1(k) \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)$ such that $H(\{j_1(k), k\}) \le b$;

take $h = j_1(k)$ in this case. Note that $h \notin \hat{J}_{\mathcal{C}}(k)$ implies that $k \notin A_h$. In either case we have $h \in \mathcal{C}$ with $H(\{h,k\}) \leq b$ and $k \notin A_h$.

In what follows, we will show that $x(A_h \cup A_k) \in F_{\mathcal{C}}$ and $x((A_h \cup A_k) \setminus \{k\}) \in F_{\mathcal{C}}$, and hence deduce that $\lambda_k = 0$.

We begin with $x(A_h \cup A_k)$. By Lemma 3, $A_h \cup A_k$ is a feasible packing, so $x(A_h \cup A_k) \in P$. Furthermore $(A_h \cup A_k) \cap \mathcal{C} = \{h\}$. Thus $|(A_h \cup A_k) \cap \mathcal{C}| = 1$, and it follows that $x(A_h \cup A_k) \in F_{\mathcal{C}}$.

We now consider $x((A_h \cup A_k) \setminus \{k\})$. Again by Lemma 3, we have that $(A_h \cup A_k) \setminus \{k\}$ is a feasible packing, and hence $x((A_h \cup A_k) \setminus \{k\}) \in P$. Furthermore, Lemma 3 shows that $((A_h \cup A_k) \setminus \{k\}) \cap \mathcal{C} = \{h\}$ also. So $|((A_h \cup A_k) \setminus \{k\}) \cap \mathcal{C}| = 1$, and $x((A_h \cup A_k) \setminus \{k\}) \in F_{\mathcal{C}}$.

Drawing these results together, we have that $\lambda x(A_h \cup A_k) = \lambda x((A_h \cup A_k) \setminus \{k\}) = \lambda_0$, and hence $0 = \lambda x(A_h \cup A_k) - \lambda x((A_h \cup A_k) \setminus \{k\}) = \lambda_k$. Since $k \in \mathcal{N} \setminus A(\mathcal{C})$ was chosen arbitrarily, it follows that $\lambda_k = 0$ for all $k \in \mathcal{N} \setminus A(\mathcal{C})$.

2. **Case 1(b)**: Let $k \in \mathcal{C}$.

   In what follows, we will show that $x(A_k) \in F_{\mathcal{C}}$ and $\lambda_j = 0$ for all $j \in A_k \setminus \{k\}$, and hence deduce that $\lambda_k = \lambda_0$. By Assumption 1 and Lemma 1(iii), A_k induces a feasible packing and so $x(A_k) \in P$.

   By Lemma 2(i) $A_k \cap \mathcal{C} = \{k\}$. Hence $|A_k \cap \mathcal{C}| = 1$, and it follows that $x(A_k) \in F_{\mathcal{C}}$. Now

$$\lambda_0 = \lambda x(A_k) = \lambda x(A_k \setminus \{k\}) + \lambda_k. \tag{6}$$

   But by Lemma 2(ii) we have $A_k \setminus \{k\} \subseteq A(\mathcal{C}) \setminus \mathcal{C}$, so from Case 1(a), $\lambda x(A_k \setminus \{k\}) = 0$. Hence (6) reduces to $\lambda_k = \lambda_0$. Since $k \in \mathcal{C}$ was chosen arbitrarily, it follows that $\lambda_k = \lambda_0$ for all $k \in \mathcal{C}$.

It has been shown that $\lambda_k = 0$ for all $k \in \mathcal{N} \setminus \mathcal{C}$ and $\lambda_k = \lambda_0$ for all $k \in \mathcal{C}$. Since we assumed that for some (λ, λ_0), $\lambda\, x = \lambda_0$ for all $x \in F_{\mathcal{C}}$, we have shown that $F_{\mathcal{C}}$ represents a facet of *conv*(P). ■

We now use Lemma 6 to show how, by the application of the following simple procedure, we can generate a maximal clique $\mathcal{C}' \subseteq \mathcal{N}$ that *does* satisfy Condition 1 from a maximal clique $\mathcal{C} \subseteq \mathcal{N}$ with $\mathcal{P}(\mathcal{C}) = \emptyset$ that does not satisfy Condition 1, and hence derive a facet-defining inequality for *conv*(P).

Procedure 1 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a maximal clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) = \emptyset$, and suppose Condition 1 does not hold. From Lemma 6 it follows that for some $k \in A(\mathcal{C}) \setminus \mathcal{C}$, $\mathcal{C}' = (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\}$ is also a maximal clique in CG with $\mathcal{P}(\mathcal{C}') = \emptyset$. Find such a k and replace $\mathcal{C}$ by $\mathcal{C}'$. Repeat until $\mathcal{C}$ satisfies Condition 1.*

Procedure 1 will always terminate with a maximal clique that satisfies Condition 1, and hence yield a clique inequality of the form (5) that defines a facet of $conv(P)$. To see that Procedure 1 terminates in less than $|A(\mathcal{C})|$ steps, we observe that since $A_k \subsetneq A(\hat{J}_{\mathcal{C}}(k))$, one has $A(\mathcal{C}') \subsetneq A(\mathcal{C})$. The following result is obvious.

Proposition 1 *Procedure 1 runs in polynomial time.*

3.2 Case 2: Non-empty intersection set, $\mathcal{P}(\mathcal{C}) \neq \emptyset$

In the case where $\mathcal{P}(\mathcal{C}) \neq \emptyset$ we are able to determine necessary and sufficient conditions under which a strengthened form of (5) is facet-defining for $conv(P)$. We also give a straightforward procedure that, given any maximal clique $\mathcal{C} \subseteq \mathcal{N}$ with $\mathcal{P}(\mathcal{C}) \neq \emptyset$, can generate a maximal clique satisfying these conditions.

We have seen that for each clique $\mathcal{C} \subseteq \mathcal{N}$ in the conflict graph CG, Lemma 5 guarantees that the corresponding clique inequality (5) is valid for P. However, in the case where $\mathcal{P}(\mathcal{C}) \neq \emptyset$, it is possible to strengthen this clique inequality as follows.

Lemma 7 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) \neq \emptyset$, and let $i \in \mathcal{P}(\mathcal{C})$. Then the inequality*

$$\sum_{j \in \mathcal{C}} x_j \;\leq\; x_i \tag{7}$$

is valid for P.

Proof. From Definition 4 we have that $i \in A_j$ for each $j \in \mathcal{C}$. It follows from the transitivity of the precedence constraints (3) that for all $j \in \mathcal{C}$, $x_j \leq x_i$. Hence if $x_i = 0$ it must be that $x_j = 0$ for all $j \in \mathcal{C}$, and (7) holds. Otherwise, $x_i = 1$ and (7) is equivalent to the clique inequality (5), which is valid for P by Lemma 5. So we have that the strengthened clique inequality (7) is valid for P when $\mathcal{P}(\mathcal{C}) \neq \emptyset$. ■

Definition 6 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) \neq \emptyset$, and let $i \in \mathcal{P}(\mathcal{C})$. Let $F^i_{\mathcal{C}} = \left\{ x \in conv(P) : \sum_{j \in \mathcal{C}} x_j \;=\; x_i \right\}$, that is, $F^i_{\mathcal{C}}$ represents the face of $conv(P)$ determined from the valid inequality (7).*

We now define $\mathcal{Q}(\mathcal{C})$ to be the set of items that (i) lie in the intersection of the entire precedence sets of all the items in the clique $\mathcal{C}$, and (ii) have no items in their successor sets D_i that satisfy the same property.

Definition 7 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a clique in the conflict graph CG. Define $\mathcal{Q}(\mathcal{C}) = \{i \in \mathcal{P}(\mathcal{C}) : \mathcal{C} \not\subseteq D_k$ for all $k \in D_i \setminus \{i\}\}$. Equivalently, $\mathcal{Q}(\mathcal{C}) = \{i \in \mathcal{P}(\mathcal{C}) : (D_i \setminus \{i\}) \cap \mathcal{P}(\mathcal{C}) = \emptyset\}$.*

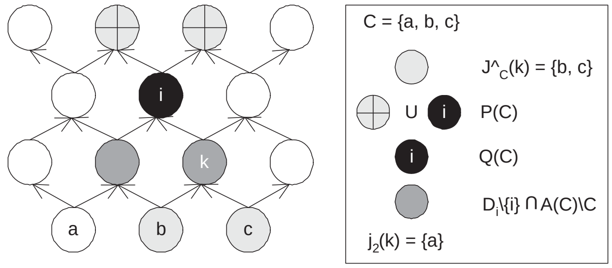


Figure 4: Illustration of the definition of $\mathcal{Q}(\mathcal{C})$ and $j_2(k)$ from Condition 2 for Case 2, $\mathcal{P}(\mathcal{C}) \neq \emptyset$, on the precedence graph $(\mathcal{N}, \mathcal{S})$. Take $a_i = 1$ for all $i \in \mathcal{N}$ and $b = 11$.

See Figure 4 for an illustration of a set $\mathcal{Q}(\mathcal{C})$. Note that if $|\mathcal{P}(\mathcal{C})| = 1$, then $\mathcal{Q}(\mathcal{C}) = \mathcal{P}(\mathcal{C})$. As we will show, the following condition is necessary and sufficient for $\mathcal{C}$ to be such that $F^i_{\mathcal{C}}$ is a facet of $conv(P)$, where $\hat{J}_{\mathcal{C}}(k)$ is defined for all $k \in A(\mathcal{C}) \setminus \mathcal{C}$ according to Definition 1.

Condition 2 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a maximal clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) \neq \emptyset$, let $\mathcal{Q}(\mathcal{C})$ be determined according to Definition 7, and let $i \in \mathcal{Q}(\mathcal{C})$. Either $(D_i \setminus \{i\}) \cap (A(\mathcal{C}) \setminus \mathcal{C}) = \emptyset$, or, for every $k \in (D_i \setminus \{i\}) \cap (A(\mathcal{C}) \setminus \mathcal{C})$, there exists $j_2(k) \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)$ such that $H(\{j, k\}) \leq b$.*

See Figure 4 for an illustration of $j_2(k)$ when $(D_i \setminus \{i\}) \cap (A(\mathcal{C}) \setminus \mathcal{C}) \neq \emptyset$.

Suppose a maximal clique $\mathcal{C} \subseteq \mathcal{N}$ in the conflict graph CG with $\mathcal{P}(\mathcal{C}) \neq \emptyset$ is given, with $\mathcal{Q}(\mathcal{C})$ determined according to Definition 7, $i \in \mathcal{Q}(\mathcal{C})$, and suppose Condition 2 does not hold. The following lemma shows that in this case (7) is redundant in the description of $conv(P)$. It also provides a way to construct another maximal clique $\mathcal{C}'$ from $\mathcal{C}$ for which Condition 2 holds.

Lemma 8 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a maximal clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) \neq \emptyset$, let $\mathcal{Q}(\mathcal{C})$ be determined according to Definition 7, and let $i \in \mathcal{Q}(\mathcal{C})$. Suppose Condition 2 does not hold; that is, $(D_i \setminus \{i\}) \cap (A(\mathcal{C}) \setminus \mathcal{C}) \neq \emptyset$ and for some $k \in (D_i \setminus \{i\}) \cap (A(\mathcal{C}) \setminus \mathcal{C})$, $H(\{j, k\}) > b$ for all $j \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)$. Then*

(i) $\mathcal{C}' = (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\}$ is also a maximal clique in CG;

(ii) $i \in \mathcal{Q}(\mathcal{C}')$, and $F^i_{\mathcal{C}} \nsubseteq F^i_{\mathcal{C}'}$; and

(iii) the strengthened clique inequality (7) for $\mathcal{C}$ and item i is redundant in the description of $conv(P)$.

Proof. We first prove part (i). From the definition of $\mathcal{Q}(\mathcal{C})$ we have that $\mathcal{C} \nsubseteq D_k$ since $k \in D_i \setminus \{i\}$. Hence $\hat{J}_{\mathcal{C}}(k) \subsetneq \mathcal{C}$. Let $\mathcal{C}' = (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\}$. Then $|\mathcal{C}'| \geq 2$ and $\mathcal{C}'$ is also a clique in $CG = (\mathcal{N}, E)$. Suppose that $\mathcal{C}'$ is not a maximal clique in CG, so there exists $m \notin \mathcal{C}'$ such that $\{m, l\} \in E$ for all $l \in \mathcal{C}'$. Note that $m \notin \hat{J}_{\mathcal{C}}(k)$ since otherwise $k \in A_m$ and $\{m, k\} \notin E$ by Assumption 1.

Since $\{m,k\} \in E$ and by Definition 1 we have that $A_k \subseteq A_h$ for all $h \in \hat{J}_{\mathcal{C}}(k)$, it follows that $\{m,h\} \in E$ for all $h \in \hat{J}_{\mathcal{C}}(k)$. So we have $\{m,l\} \in E$ for all $l \in \mathcal{C}' \cup \hat{J}_{\mathcal{C}}(k) = (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\} \cup \hat{J}_{\mathcal{C}}(k)$. In particular $\{m,l\} \in E$ for all $l \in \mathcal{C}$, which contradicts the maximality of $\mathcal{C}$. Hence $\mathcal{C}' = (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\}$ is also a maximal clique in CG.

We now prove part (ii). First, we show that $i \in P(\mathcal{C}')$. To begin, $i \in P(\mathcal{C})$ so $i \in A_j$ for all $j \in \mathcal{C}$, and hence for all $j \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)$. Furthermore $k \in D_i$ so $i \in A_k$ by the definition of descendent sets. Thus $i \in A_j$ for all $j \in \mathcal{C}'$, i.e. $i \in P(\mathcal{C}')$. Now suppose that $i \notin \mathcal{Q}(\mathcal{C}')$. Then there must exist $h \in D_i \setminus \{i\}$ such that $D_h \supseteq \mathcal{C}'$. But since $i \in \mathcal{Q}(\mathcal{C})$, we know that $D_h \not\supseteq \mathcal{C}$, so it must be that $D_h \not\supseteq \hat{J}_{\mathcal{C}}(k)$. Now $k \in \mathcal{C}' \subseteq D_h$ so $D_k \subseteq D_h$ by Property 1 and the definition of descendent sets. But $\hat{J}_{\mathcal{C}}(k) \subseteq D_k$, which contradicts $D_h \not\supseteq \hat{J}_{\mathcal{C}}(k)$. Thus it must be that $i \in \mathcal{Q}(\mathcal{C}')$ as required.

Let $H^i_{\mathcal{C}} = \{x \in \mathbb{R}^{|\mathcal{N}|} : \sum_{j \in \mathcal{C}} x_j = x_i\}$ and $H^i_{\mathcal{C}'} = \{x \in \mathbb{R}^{|\mathcal{N}|} : \sum_{j \in \mathcal{C}'} x_j = x_i\}$. Let $I^i_{\mathcal{C}} = P \cap H^i_{\mathcal{C}}$ and $I^i_{\mathcal{C}'} = P \cap H^i_{\mathcal{C}'}$. We will show that $I^i_{\mathcal{C}} \subsetneq I^i_{\mathcal{C}'}$. Let $x \in I^i_{\mathcal{C}}$. Then either $x_i = 0$ or $x_i = 1$. Consider first $x_i = 0$. By the validity of (7) $x_j = 0$ for all $j \in \mathcal{C}$. Similarly, since $k \in D_i$ it follows that $i \in A_k$ and hence $x_k = 0$. Thus $x \in I^i_{\mathcal{C}'}$. Now consider the case $x_i = 1$. Then there is exactly one $j \in \mathcal{C}$ such that $x_j = 1$. If $j \in \hat{J}_{\mathcal{C}}(k)$ then $k \in A_j$ and $x_k = 1$, implying that $x \in I^i_{\mathcal{C}'}$. If $j \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)$ then obviously $x \in I^i_{\mathcal{C}'}$. Hence $I^i_{\mathcal{C}} \subseteq I^i_{\mathcal{C}'}$. Now consider the feasible packing A_k. Since $k \in D_i$, we have that $i \in A_k$ and hence $x(A_k) \in I^i_{\mathcal{C}'}$. However, since $k \in A(\mathcal{C}) \setminus \mathcal{C}$, by Lemma 2(iii) $A_k \subseteq A(\mathcal{C}) \setminus \mathcal{C}$, and hence $A_k \cap \mathcal{C} = \emptyset$. Thus $x(A_k) \notin I^i_{\mathcal{C}}$. Hence $I^i_{\mathcal{C}} \subsetneq I^i_{\mathcal{C}'}$. Since $I^i_{\mathcal{C}}$ and $I^i_{\mathcal{C}'}$ are sets of binary vectors, it follows that $conv(I^i_{\mathcal{C}}) \subsetneq conv(I^i_{\mathcal{C}'})$.

Now observe that $F^i_{\mathcal{C}} = conv(P) \cap H^i_{\mathcal{C}}$ and $F^i_{\mathcal{C}'} = conv(P) \cap H^i_{\mathcal{C}'}$, and furthermore by Lemma 7 the hyperplanes $H^i_{\mathcal{C}}$ and $H^i_{\mathcal{C}'}$ are defined by *valid* inequalities for P. Thus by Lemma 6.1.1 of Balas [2] we have that $conv(I^i_{\mathcal{C}}) = conv(P \cap H^i_{\mathcal{C}}) = conv(P) \cap H^i_{\mathcal{C}} = F^i_{\mathcal{C}}$, and that $conv(I^i_{\mathcal{C}'}) = conv(P \cap H^i_{\mathcal{C}'}) = conv(P) \cap H^i_{\mathcal{C}'} = F^i_{\mathcal{C}'}$, and hence $F^i_{\mathcal{C}} \subsetneq F^i_{\mathcal{C}'}$.

Finally, we prove part (iii). Consider the feasible packing A_i. Then $x(A_i) \in P$ but since $j \in D_i$ for all $j \in \mathcal{C}'$, $x(A_i) \notin F^i_{\mathcal{C}'}$, and hence $F^i_{\mathcal{C}'} \subsetneq P$. It follows from Lemma 4 that $F^i_{\mathcal{C}}$ cannot represent a facet of $conv(P)$, and hence the clique inequality (7) for $\mathcal{C}$ and item i is redundant in the description of $conv(P)$. ■

As a consequence of Lemma 8, Condition 2 is a necessary condition for the inequality (7) to be facet defining. The next theorem states that this condition is also sufficient.

Theorem 2 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a maximal clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) \neq \emptyset$, let $\mathcal{Q}(\mathcal{C})$ be determined according to Definition 7, and let $i \in \mathcal{Q}(\mathcal{C})$. Then $F^i_{\mathcal{C}}$ from Definition 6 is a facet of $conv(P)$ if and only if Condition 2 holds.*

Proof.

($\Rightarrow$) This is a direct consequence of Lemma 8.

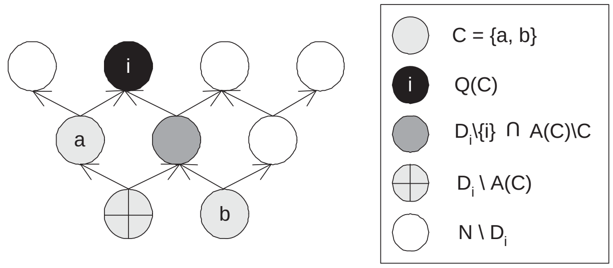


Figure 5: Illustration of the situations for consideration in Case 2, $\mathcal{P}(\mathcal{C}) \neq \emptyset$.

($\Leftarrow$) Consider $\mathcal{C} \subseteq \mathcal{N}$ such that $\mathcal{C}$ is a maximal clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) \neq \emptyset$, $\mathcal{Q}(\mathcal{C})$ is defined as in Definition 7, $i \in \mathcal{Q}(\mathcal{C})$, and Condition 2 holds. Suppose that $\lambda\, x = \lambda_0$ for all $x \in F^i_{\mathcal{C}}$, where $F^i_{\mathcal{C}}$ is given by Definition 6. Note that the zero vector induces a feasible packing, since it is always feasible to have an empty knapsack. Hence $0 \in P$. In this case, $0 \in F^i_{\mathcal{C}}$ as well, and hence $\lambda_0 = 0$. Thus if it can be shown that $\lambda_k = -\lambda_i$ for all $k \in \mathcal{C}$, and $\lambda_k = 0$ for all $k \in \mathcal{N} \setminus (\mathcal{C} \cup \{i\})$, then by Theorem 3.6 of Nemhauser and Wolsey [12] we will have proved that $F^i_{\mathcal{C}}$ represents a facet of $conv(P)$. There are three cases to consider, described in detail below and illustrated in Figure 5.

1. **Case 2(a)**: Let $k \in \mathcal{N} \setminus D_i$.

   In what follows, we will show that $x(A_k) \in F^i_{\mathcal{C}}$ and $x(A_k \setminus \{k\}) \in F^i_{\mathcal{C}}$, and hence deduce that $\lambda_k = 0$. We begin by considering $x(A_k)$. By Lemma 1(iii) A_k is a feasible packing and so $x(A_k) \in P$. Since $k \notin D_i$, by Lemma 1(ii) we have that $A_k \subseteq \mathcal{N} \setminus D_i$. From the definition of $\mathcal{Q}(\mathcal{C})$ we also have that $\mathcal{C} \subseteq D_i$, and thus $A_k \cap \mathcal{C} = \emptyset$. From the definition of D_i it follows that since $k \notin D_i$, $i \notin A_k$, and we have $x(A_k) \in F^i_{\mathcal{C}}$.

   We now consider $x(A_k \setminus \{k\})$. By Lemma 1 (v) we have that $A_k \setminus \{k\}$ is a feasible packing. Hence $x(A_k \setminus \{k\}) \in P$. From above we have that $A_k \cap \mathcal{C} = \emptyset$, and it follows directly that $(A_k \setminus \{k\}) \cap \mathcal{C} = \emptyset$. Since $i \notin A_k$ we also have that $i \notin A_k \setminus \{k\}$. Hence $x(A_k \setminus \{k\}) \in F^i_{\mathcal{C}}$ also.

   Drawing these results together, we have that $\lambda x(A_k) = \lambda x(A_k \setminus \{k\}) = \lambda_0$, and hence $0 = \lambda x(A_k) - \lambda x(A_k \setminus \{k\}) = \lambda_k$. Since $k \in \mathcal{N} \setminus D_i$ was chosen arbitrarily, it follows that $\lambda_k = 0$ for all $k \in \mathcal{N} \setminus D_i$.

2. **Case 2(b)**: Let $k \in (D_i \setminus \{i\}) \setminus \mathcal{C}$.

   There are two subcases: $k \in A(\mathcal{C})$ and $k \notin A(\mathcal{C})$. In the first case, by Condition 2 there exists $j_2(k) \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)$ such that $H(\{j_2(k), k\}) \leq b$; take $h = j_2(k)$ in this case. Since $h \notin \hat{J}_{\mathcal{C}}(k)$, it must be that $k \notin A_h$. In the second case, since $\mathcal{C}$ is maximal there must exist at least one $h \in \mathcal{C}$ such that $H(\{h, k\}) \leq b$. In this it must obviously be that $k \notin A_h$, since $h \in \mathcal{C}$ and $k \notin A(\mathcal{C})$. In either case, we have $h \in \mathcal{C}$ with $H(\{h, k\}) \leq b$ and $k \notin A_h$.

Now by Lemma 3, it must be that $x(A_h \cup A_k) \in P$, and $(A_h \cup A_k) \cap \mathcal{C} = \{h\}$. Furthermore $k \in D_i$ so $i \in A_k$, thus $x(A_h \cup A_k) \in F^i_{\mathcal{C}}$. Also by Lemma 3, it must be that $x((A_h \cup A_k) \setminus \{k\}) \in P$, and $((A_h \cup A_k) \setminus \{k\}) \cap \mathcal{C} = \{h\}$. Furthermore $k \in D_i \setminus \{i\}$ so again $i \in A_k$ and $x((A_h \cup A_k) \setminus \{k\}) \in F^i_{\mathcal{C}}$.

Thus $\lambda x(A_h \cup A_k) = \lambda x((A_h \cup A_k) \setminus \{k\}) = \lambda_0$, and hence $0 = \lambda x(A_h \cup A_k) - \lambda x((A_h \cup A_k) \setminus \{k\}) = \lambda_k$. Since $k \in (D_i \setminus \{i\}) \setminus \mathcal{C}$ was chosen arbitrarily, it follows that $\lambda_k = 0$ for all $k \in (D_i \setminus \{i\}) \setminus \mathcal{C}$.

3. **Case 2(c)**: Let $k \in \mathcal{C}$.

   In what follows, we will show that $x(A_k) \in F^i_{\mathcal{C}}$ and $\lambda_j = 0$ for all $j \in A(\mathcal{C}) \setminus (\mathcal{C} \cup \{i\})$, and hence deduce that $\lambda_k = -\lambda_i$. By Lemma 1(iii), A_k induces a feasible packing and we have $x(A_k) \in P$. Since $k \in \mathcal{C}$, from Lemma 2(i) $A_k \cap \mathcal{C} = \{k\}$, and $|A_k \cap \mathcal{C}| = 1$. By the definition of $\mathcal{Q}(\mathcal{C})$, $\mathcal{C} \subseteq D_i$ and thus $k \in D_i$, hence $i \in A_k$, and we have $x(A_k) \in F^i_{\mathcal{C}}$.

   By Lemma 2(ii) we have $A_k \setminus \{k\} \subseteq A(\mathcal{C}) \setminus \mathcal{C}$. From Case 2(a) we have that $\lambda_j = 0$ for all $j \in \mathcal{N} \setminus D_i$ and from Case 2(b) we have that $\lambda_j = 0$ for all $j \in (D_i \setminus \{i\}) \setminus \mathcal{C}$. Hence $\lambda_j = 0$ for all $j \in A(\mathcal{C}) \setminus (\mathcal{C} \cup \{i\})$, and so $\lambda x(A_k) = \lambda_k + \lambda_i = \lambda_0$. Thus $\lambda_k = -\lambda_i$. Since $k \in \mathcal{C}$ was chosen arbitrarily, it follows that $\lambda_k = -\lambda_i$ for all $k \in \mathcal{C}$.

It has been shown that $\lambda_k = 0$ for all $k \in \mathcal{N} \setminus (\mathcal{C} \cup \{i\})$ and $\lambda_k = -\lambda_i$ for all $k \in \mathcal{C}$. Since we assumed that for some (λ, λ_0), $\lambda\, x = \lambda_0$ for all $x \in F^i_{\mathcal{C}}$, we have shown that $F^i_{\mathcal{C}}$ represents a facet of $conv(P)$. ■

We now use Lemma 8 to show how, by the application of the following simple procedure, we can generate a maximal clique $\mathcal{C}' \subseteq \mathcal{N}$ that *does* satisfy Condition 2 from a maximal clique $\mathcal{C} \subseteq \mathcal{N}$ that does not satisfy Condition 2, and hence derive a facet-defining inequality for $conv(P)$.

Procedure 2 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a maximal clique in the conflict graph CG with $\mathcal{P}(\mathcal{C}) \neq \emptyset$, let $\mathcal{Q}(\mathcal{C})$ be determined according to Definition 7, and let $i \in \mathcal{Q}(\mathcal{C})$. Suppose Condition 2 does not hold. From Lemma 8 it follows that for some $k \in (A(\mathcal{C}) \setminus \mathcal{C}) \cap (D_i \setminus \{i\})$, $\mathcal{C}' = (\mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)) \cup \{k\}$ is also a maximal clique in CG. Find such a k and replace $\mathcal{C}$ by $\mathcal{C}'$. Repeat this procedure until $\mathcal{C}$ satisfies Condition 2.*

Using the same argument as we did for Procedure 1, we see that Procedure 2 will always terminate in less than $|A(\mathcal{C})|$ steps with a maximal clique that satisfies Condition 2. Upon termination we have a clique inequality of the form (7) that defines a facet of $conv(P)$. The following result is obvious.

Proposition 2 *Procedure 2 runs in polynomial time.*

We now consider the polyhedral investigations of the PCKP carried out by other authors, and give a comparison of the different approaches.

4 Cover-Based Polyhedral Approaches to the PCKP

Investigation of the polyhedral structure of the PCKP has previously been carried out by Boyd [6], Park and Park [13] and van de Leensel *et al.* [16]. All of these authors consider the derivation of strong inequalities for P by applying lifting procedures to valid knapsack cover-based inequalities. We now consider some of the knapsack cover-based inequalities studied by these authors, and compare them to the clique inequalities introduced in Section 3. We begin by giving a summary of the relevant PCKP terminology and the main cover-based results.

4.1 PCKP Terminology and Cover-Based Results

The basic terminology for the PCKP used by Boyd [6], Park and Park [13] and van de Leensel *et al.* [16] is as follows. Two items $i,\ j \in \mathcal{N}$ are called **incomparable** if $i \notin A_j$ and $j \notin A_i$. A set $B \subseteq \mathcal{N}$ is incomparable if the elements of B are pairwise incomparable. An incomparable set $C \subseteq \mathcal{N}$ is an **induced cover** if $H(C) > b$. It is obvious that for any induced cover $C \subseteq \mathcal{N}$ the inequality

$$\sum_{j \in C} x_j \ \leq \ |C| - 1 \tag{8}$$

is valid for P.

Park and Park [13] call an induced cover a **minimal induced cover (MIC)** if $H(C \setminus \{i\}) \leq b$ for all $i \in C$, in which case the associated cover inequality (8) is called an **MIC inequality**. This definition differs from that used by Boyd [6] and van de Leensel *et al.* [16]. Boyd [6] defines an induced cover $C \subseteq \mathcal{N}$ to be minimal if C is incomparable and $H(C) - a_i \leq b$ for all $i \in \mathcal{C}$. We shall call such an induced cover a **Boyd minimal cover (BMC)**, and the associated cover inequality a **BMC inequality**. As Park and Park [13] note, in general, a BMC is also an MIC, but the converse does not hold.

Boyd [6] and van de Leensel *et al.* [16] also consider a generalized version of a Boyd minimal cover. We define a **K-Boyd minimal cover (K-BMC)** to be an induced cover $C \subseteq \mathcal{N}$ with the property that for all $B \subseteq C$ with $|B| = K$, B is a BMC. It is obvious that if C is a K-BMC then

$$\sum_{j \in C} x_j \ \leq \ K - 1 \tag{9}$$

is valid for P. Note that C a $|C|$-BMC is simply a BMC. Boyd [6] and van de Leensel *et al.* [16] do not seek results for inequalities of the form (9) under looser conditions.

Van de Leensel *et al.* [16] are the only authors known to us who show how to derive facet-defining inequalities for $conv(P)$. They redefine the term minimal induced cover to coincide with the definition of a Boyd minimal cover, and consider BMCs (which they call MICs) in their study of the PCKP polyhedron P. In the sequel we will continue to distinguish between MICs and BMCs.

Van de Leensel *et al.* [16] consider BMCs and K-BMCs separately, and for both cases develop a general sequential lifting procedure to lift the (K-)cover inequality to a facet of $conv(P)$. The resulting facet-defining inequality takes the form

$$\sum_{i \in C} x_i \; + \sum_{i \in A(C) \setminus C} \alpha_i (1 - x_i) \; + \sum_{i \in \mathcal{N} \setminus A(C)} \alpha_i x_i \; \leq \; K - 1, \qquad (10)$$

where of course $K = |C|$ in the case of a BMC.

They note that in general, calculating each lifting coefficient α_i requires the solution of a separate PCKP. Further, they show that determining the maximal lifting coefficients for the items in the set $\mathcal{N} \setminus A(C)$ is NP-complete in the strong sense. Finally, a polynomial time algorithm for determining the maximal lifting coefficients for items in the set $A(C) \setminus C$ is presented. This algorithm applies only in the special case $K = |C|$. We will use it to determine strengthened BMC inequalities in examples in Section 5.

Park and Park [13] consider MICs $C \subseteq \mathcal{N}$, and present a heuristic for determining lifting coefficients for items in the set $A(C) \setminus C$ to strengthen the MIC inequality (8). They show that under certain conditions, this lifted inequality is facet-defining for the lower-dimensional polyhedron $P(C)$, defined as $P(C) = conv(proj_{A(C)}\{x(D) \in P : D \subseteq A(C)\})$ for any incomparable set $C \subseteq \mathcal{N}$. That is, $P(C)$ is the convex hull of P restricted to those variables in $A(C)$. No further significant results for MICs have been developed.

For our investigation, we extend the concept of an MIC in a similar manner to that used by Boyd [6] and van de Leensel *et al.* [16] for BMCs. A set $C \subseteq \mathcal{N}$ is a K**-MIC** if C is incomparable, and for all $B \subseteq C$ with $|B| = K$, B is an MIC, in which case we call the corresponding K-cover inequality (9) a K**-MIC inequality**. As in the case of BMCs, MICs are special cases of K-MICs. Hence we consider only K-BMCs and K-MICs in the sequel. Note that the concept of a K-MIC has not been investigated by any previous author.

4.2 Comparison of Covers and Cliques in the Conflict Graph

Consider an instance of PCKP and let $CG = (\mathcal{N}, E)$ be a conflict graph determined according to Definition 3. We now compare the covers investigated by Boyd [6], Park and Park [13] and van de Leensel *et al.* [16] with cliques in the conflict graph CG. Of particular interest in our investigation are 2-BMCs and 2-MICs: there is a precise correspondence between 2-MICs and cliques in the conflict graph CG. The following results are straightforward to prove.

Lemma 9 *Let $\mathcal{C} \subseteq \mathcal{N}$.*

(i) $\mathcal{C}$ is a 2-MIC if and only if $\mathcal{C}$ is a clique in the conflict graph.

(ii) If $\mathcal{C}$ is a K-BMC then $\mathcal{C}$ is a K-MIC.

(iii) If $\mathcal{C}$ is a K-MIC then $\mathcal{C}$ is not a k-MIC for any $k \neq K$.

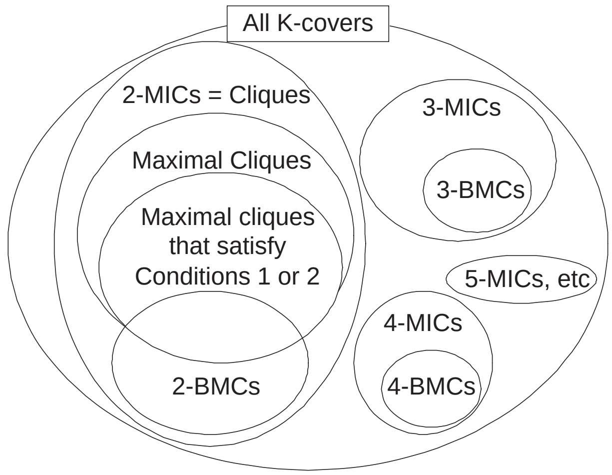


Figure 6: Diagram of the set of all K-covers

Corollary 1 *Let $\mathcal{C} \subseteq \mathcal{N}$. If $\mathcal{C}$ is a clique in the conflict graph then*

*(i) $\mathcal{C}$ **cannot** be a K-MIC where $K > 2$; and*

*(ii) $\mathcal{C}$ **cannot** be a K-BMC where $K > 2$.*

We use Lemma 9 and Corollary 1 to demonstrate where cliques determined from the conflict graph CG fit into the set of all K-covers. These results, along with the existence of 2-BMCs that are not maximal cliques (see the example described in Figure 9), justify everything in Figure 6 except for the placement of the set of maximal cliques that satisfy Conditions 1 or 2. In fact, this, too, is justified, as we will now show.

Lemma 10 *Let $\mathcal{C} \subseteq \mathcal{N}$ be a maximal clique in the conflict graph CG. If $\mathcal{C}$ is also a 2-BMC, then either $\mathcal{P}(\mathcal{C}) = \emptyset$ and Condition 1 is satisfied, or $\mathcal{P}(\mathcal{C}) \neq \emptyset$ and Condition 2 is satisfied.*

Proof. Let $\mathcal{C} \subseteq \mathcal{N}$ be a maximal clique in the conflict graph CG, and suppose $\mathcal{C}$ is also a 2-BMC. We begin by showing that If $\mathcal{P}(\mathcal{C}) = \emptyset$, then suppose $A(\mathcal{C}) \setminus \mathcal{C} \neq \emptyset$ (otherwise Condition 1 holds), and let $k \in A(\mathcal{C}) \setminus \mathcal{C}$. It must be that $\hat{J}_{\mathcal{C}}(k) \neq \mathcal{C}$, otherwise $k \in \mathcal{P}(\mathcal{C})$, which is a contradiction.

If $\mathcal{P}(\mathcal{C}) \neq \emptyset$, let $i \in \mathcal{Q}(\mathcal{C})$, suppose $(D_i \setminus \{i\}) \cap (A(\mathcal{C}) \setminus \mathcal{C}) \neq \emptyset$ (otherwise Condition 2 holds), and let $k \in (D_i \setminus \{i\}) \cap (A(\mathcal{C}) \setminus \mathcal{C})$. It must be that $\hat{J}_{\mathcal{C}}(k) \neq \mathcal{C}$, otherwise $k \in \mathcal{P}(\mathcal{C})$ and $\mathcal{C} \subseteq D_k$, which is a contradiction of the assumption that $i \in \mathcal{Q}(\mathcal{C})$.

In either case, choose any $j \in \mathcal{C} \setminus \hat{J}_{\mathcal{C}}(k)$, and also choose any $m \in \hat{J}_{\mathcal{C}}(k)$. Now $k \in A_m$, so by Property 1, $A_k \subseteq A_m$, and hence $A(\{j,k\}) \subseteq A(\{j,m\})$. Also $m \notin A_k$, since $A_k \subseteq A(\mathcal{C}) \setminus \mathcal{C}$ by Lemma 1 (i), and $m \in \hat{J}_{\mathcal{C}}(k) \subseteq \mathcal{C}$. Furthermore, $m, j \in \mathcal{C}$ so $m \notin A_j$. Thus $m \notin A_j \cup A_k = A(\{j,k\})$. It follows that $A(\{j,k\}) \subseteq A(\{j,m\}) \setminus \{m\}$. Since $\mathcal{C}$ is a 2-BMC and $j, m \in \mathcal{C}$ we have $H(A(\{j,m\}) \setminus \{m\}) = H(\{j,m\}) - a_m \leq b$ and hence $H(\{j,k\}) \leq b$.

In the case $\mathcal{P}(\mathcal{C}) = \emptyset$, this shows that Condition 1 is satisfied; otherwise it shows that Condition 2 is satisfied. ∎

To complete the classification given in Figure 6, and highlight the new contribution of clique-based inequalities, we note that there *do* exist examples of maximal cliques satisfying Conditions 1 or 2 that are *not* 2-BMCs: Example 1 in the following section has *no* 2-BMCs, but has four maximal cliques in the conflict graph satisfying either Condition 1 or Condition 2.

It is not hard to see from the form of (10) that it is impossible to arrive at an inequality of the form of either (5) or (7) by maximal lifting, unless $K = 2$. Of course, it is certainly possible that our facet-defining clique-based inequalities could be derived by maximal lifting of 2-MIC inequalities. However, as van de Leensel *et al.* [16] show, maximal lifting may require the solution of NP-complete subproblems; their polynomial time lifting algorithm only applies to the special case that $K = |C|$ and only to coefficients of variables in $A(C) \setminus C$, not to the whole of $\mathcal{N}$. Furthermore, van de Leensel *et al.* [16] do not discuss the facet-defining status of (9) in the case where C is a 2-MIC rather than 2-BMC.

Finally, we note that the clique inequalities are very attractive from a practical point of view. In order to obtain a single facet defining inequality via the sequential lifting approach proposed by van de Leensel *et al.* [16], one has to solve numerous lifting problems, which are computationally demanding even in polynomially solvable special cases. The problem of finding a violated clique inequality, on the other hand, boils down to the solution of only one maximum weight clique problem. This problem is NP-hard as well, but there are numerous very efficient heuristics and exact solution methods for finding maximum weight cliques in general graphs; see Bomze *et al.* [4] for example. Furthermore, efficient implementations of these algorithms already exist in all state-of-the-art integer programming solvers for the generation of general clique inequalities. Applying these clique algorithms to the conflict graph defined by a PCK (sub-)problem, the clique-based facets described in Section 3 thus can be found very efficiently in practice.

5 Application of Clique-Based Inequalities to PCKP Examples

We now demonstrate that our clique-based approach to determining facets of $conv(P)$ can find facets that would not be found using the cover-based approaches of previous authors, and demonstrate their relative strengthening effect on the LP relaxation. To be fair, we restrict our attention to polynomial time approaches, and do not attempt any lifting except that for which polynomial time algorithms have been developed.

We give two PCKP examples. We find all K-BMCs, and where applicable,

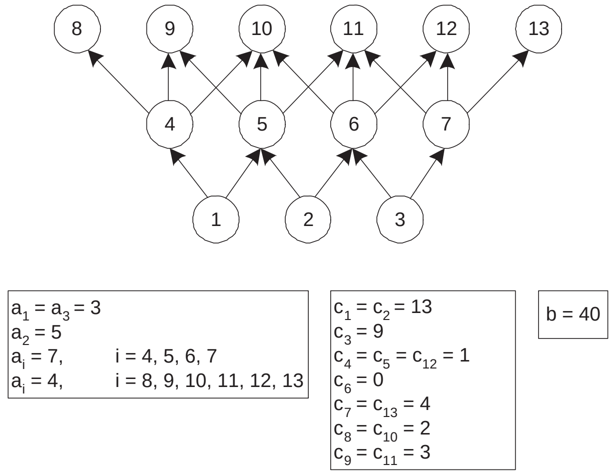


Figure 7: PCKP Example 1.

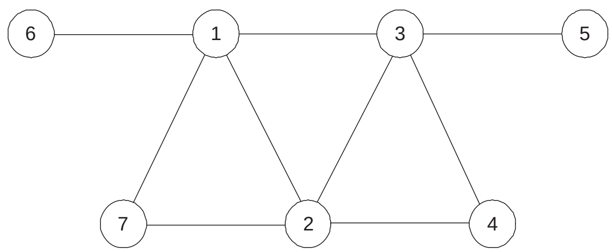


Figure 8: Conflict Graph for PCKP Example 1.

use the polynomial time algorithm of van de Leensel *et al.* [16] and the procedure of Park and Park [13] to lift these. We also find all maximal cliques in the conflict graph, and apply Procedures 1 or 2 as appropriate to derive all facet-defining clique-based inequalities.

Consider Example 1 given in Figure 7. There are eight K-BMCs in this example, all of which are 3-BMCs. Note in particular that there are no 2-BMCs in this example. For the 3-BMCs with $|C| = 3$, for which $A(C) \setminus C \neq \emptyset$, we apply the polynomial time lifting algorithm of van de Leensel *et al.* [16] to the items in the predecessor sets $A(C) \setminus C$ to strengthen the 3-BMC inequalities of the form (9). The 3-BMCs and corresponding inequalities are given in Table 1.

Of course if lifting for K-BMCs with $|C| > K$ were available, it would also be possible to strengthen the inequality for the cover $\{4, 5, 6, 7\}$: this could be lifted to either $x_4 + x_5 + x_6 + x_7 \leq x_{10} + 1$, or $x_4 + x_5 + x_6 + x_7 \leq x_{11} + 1$. For interest, we tried adding these to the LP relaxation; we found doing so did not change the value reported in Table 5.

Applying the approach for deriving facets of $conv(P)$ from clique inequalities to this example, we obtain the conflict graph given in Figure 8. Note that each

Table 1: K-BMCs for PCKP Example 1

K-BMC	Corresponding K-BMC inequality (lifted when $A(C) \setminus C \neq \emptyset$ and $K = \|C\|$)
$\{4,5,6\}$	$x_4 + x_5 + x_6 \leq x_9 + x_{10}$
	$x_4 + x_5 + x_6 \leq x_9 + x_{11}$
	$x_4 + x_5 + x_6 \leq x_{10} + x_{11}$
$\{5,6,7\}$	$x_5 + x_6 + x_7 \leq x_{10} + x_{11}$
	$x_5 + x_6 + x_7 \leq x_{10} + x_{12}$
	$x_5 + x_6 + x_7 \leq x_{11} + x_{12}$
$\{4,5,7\}$	$x_4 + x_5 + x_7 \leq x_9 + x_{11}$
	$x_4 + x_5 + x_7 \leq x_{10} + x_{11}$
$\{4,6,7\}$	$x_4 + x_6 + x_7 \leq x_{10} + x_{11}$
	$x_4 + x_6 + x_7 \leq x_{10} + x_{12}$
$\{4,5,6,7\}$	$x_4 + x_5 + x_6 + x_7 \leq 2$
$\{1,12,13\}$	$x_1 + x_{12} + x_{13} \leq 2$
$\{3,8,9\}$	$x_3 + x_8 + x_9 \leq 2$
$\{2,8,13\}$	$x_2 + x_8 + x_{13} \leq 2$

Table 2: Maximal Cliques for PCKP Example 1

Maximal Clique	Corresponding facet-defining clique inequality
$\{1,6\}$	$x_1 + x_6 \leq x_{10}$
	$x_1 + x_6 \leq x_{11}$
$\{3,5\}$	$x_3 + x_5 \leq x_{10}$
	$x_3 + x_5 \leq x_{11}$
$\{1,2,3\}$	Not facet-defining
$\{1,2,7\}$	$x_1 + x_2 + x_7 \leq x_{11}$
$\{2,3,4\}$	$x_2 + x_3 + x_4 \leq x_{10}$

clique in this conflict graph represents a 2-MIC that is not a 2-BMC. There are five maximal cliques $\mathcal{C} \subseteq \mathcal{N}$ in this conflict graph as shown in Table 2, all of which are such that $\mathcal{P}(\mathcal{C}) \neq \emptyset$. One of the maximal cliques ($\mathcal{C}^3 = \{1,2,3\}$) does not satisfy Condition 2. However, a maximal clique that does satisfy Condition 2 can be derived from $\mathcal{C}^3$ by the application of Procedure 2. For example, by replacing items 1 and 2 in $\mathcal{C}^3$ with their immediate predecessor item 5, we obtain $\mathcal{C}^2 = \{3,5\}$, which does satisfy Condition 2.

The maximal cliques and the corresponding facet-defining strengthened clique inequalities are given in Table 2. Note that none of the six facet-defining inequalities derived from these maximal cliques appear in Table 1. This must be the case since there were no 2-BMCs in this example (cf. discussion in Section 4.2). Hence we see that the clique-based approach has derived facets of $conv(P)$ that *cannot be found* using the polynomial time cover-based approach of previous authors.

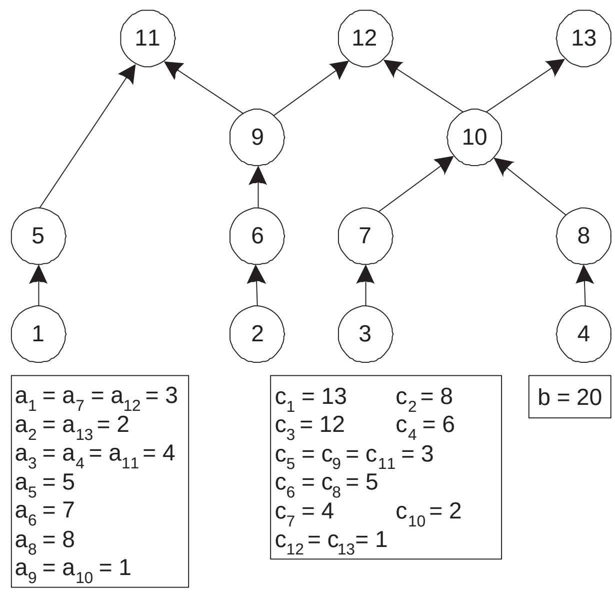


Figure 9: PCKP Example 2.

Consider now Example 2 given in Figure 9. There are nine K-BMCs $C \subseteq \mathcal{N}$ in this example, all of which are 2-BMCs. For the 2-BMCs with $|C| = 2$ for which $A(C) \setminus C \neq \emptyset$, application of the polynomial time lifting algorithm of van de Leensel *et al.* [16] for the items in the predecessor sets $A(C) \setminus C$ allows us to strengthen the 2-BMC inequalities of the form (9). The 2-BMCs and corresponding inequalities are given in Table 3.

Applying the approach for deriving facets of $conv(P)$ from clique inequalities to this example, we obtain the conflict graph given in Figure 10. Again note that each clique in this conflict graph represents a 2-MIC; only $\{4, 11\}$ is a 2-BMC. There are ten maximal cliques $\mathcal{C} \subseteq \mathcal{N}$ in this conflict graph, as shown in Table 4, all of which are such that $\mathcal{P}(\mathcal{C}) = \emptyset$, and six of which do not satisfy Condition 1. However, a maximal clique that does satisfy Condition 1 can be derived from these maximal cliques in all instances, by the application of Procedure 1. As a result, there are four clique-based inequalities that are facet-defining for this example, as seen in Table 4. In this case, the maximal clique $\{4, 11\}$ is also a 2-BMC, and since it satisfies Condition 1, we see that Lemma 10 holds. The remaining maximal cliques in the conflict graph CG all contain 2-BMCs within them, and we see that all 2-BMCs are cliques in the conflict graph, but not necessarily maximal cliques. In this case the facet-defining clique-based inequalities could be reproduced by the lifting approach of van de Leensel *et al.* [16], but in all cases some of the lifted variables lie in the set $\mathcal{N} \setminus A(C)$, and so would require solution of a difficult lifting problem. Using maximal cliques in the conflict graph CG has bypassed the need to solve difficult lifting problems.

A comparison of the LP-relaxations for the PCKP Examples 1 and 2 is

Table 3: K-BMCs for PCKP Example 2

K-BMC	Corresponding K-BMC inequality (lifted when $A(C) \setminus C \neq \emptyset$ and $K = \lvert C \rvert$)
$\{1,7\}$	$x_1 + x_7 \leq 1$
$\{3,5\}$	$x_3 + x_5 \leq 1$
$\{3,8\}$	$x_3 + x_8 \leq x_{10}$
$\{4,7\}$	$x_4 + x_7 \leq x_{10}$
$\{4,11\}$	$x_4 + x_{11} \leq 1$
$\{5,8\}$	$x_5 + x_8 \leq 1$
$\{6,7\}$	$x_6 + x_7 \leq x_{12}$
$\{6,8\}$	$x_6 + x_8 \leq x_{12}$
$\{3,5,8\}$	$x_3 + x_5 + x_8 \leq 1$

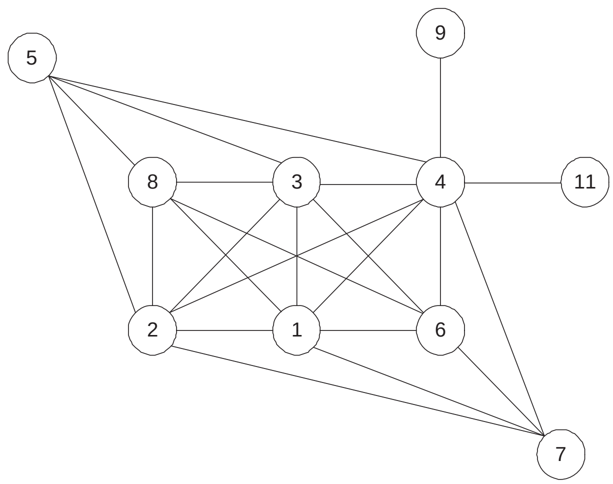


Figure 10: Conflict Graph for PCKP Example 2.

presented in Table 5. The cases tested are those of the standard integer programming formulation (PCKP), and this formulation with the addition of the K-BMC inequalities (lifted on their predecessor variables where possible), and also with the addition of the facet-defining clique inequalities. It is evident from Table 5 that the addition of the K-BMC inequalities, lifted on their predecessor variables where possible, results in a reduction in the root node gap (of approximately 6% in example 1 and 10% in example 2). The addition of the facet-defining clique-based inequalities to the PCKP formulation results in a further reduction in root node gap (of approximately 15% in both cases). In the second example the optimal integer solution is found by solving the LP-relaxation of (PCKP) with the addition of the facet-defining clique inequalities. These results indicate that the addition of facet-defining clique-based inequali-

Table 4: Maximal Cliques for PCKP Example 2

Maximal Clique	Corresponding facet-defining clique inequality
$\{1, 2, 3, 4\}$	Not facet-defining
$\{1, 2, 3, 8\}$	Not facet-defining
$\{1, 3, 4, 6\}$	Not facet-defining
$\{1, 3, 6, 8\}$	$x_1 + x_3 + x_6 + x_8 \leq 1$
$\{1, 2, 4, 7\}$	Not facet-defining
$\{1, 4, 6, 7\}$	$x_1 + x_4 + x_6 + x_7 \leq 1$
$\{2, 3, 4, 5\}$	Not facet-defining
$\{2, 3, 5, 8\}$	$x_2 + x_3 + x_5 + x_8 \leq 1$
$\{4, 9\}$	Not facet-defining
$\{4, 11\}$	$x_4 + x_{11} \leq 1$

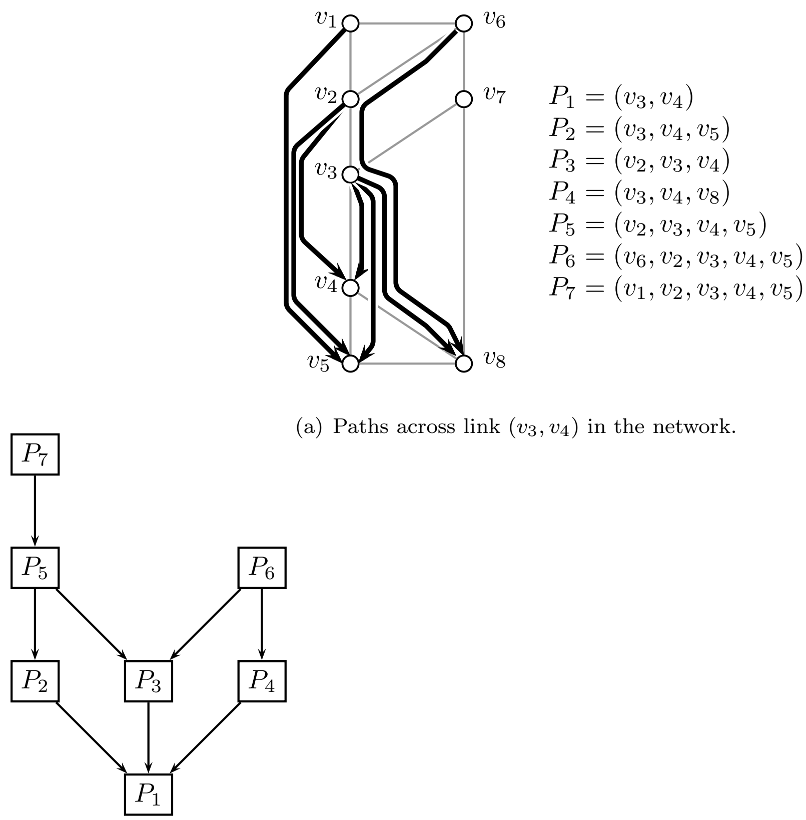
Table 5: Summary of Results for PCKP Examples

Example Number	Formulation	LP relaxation	IP value	Gap (%)
1	PCKP formulation only	35.73	29.00	23.20
	PCKP formulation with 3-BMC inequalities	34.00	29.00	17.24
	PCKP formulation and facet-defining clique inequalities	29.75	29.00	2.59
2	PCKP formulation only	32.31	26.00	24.26
	PCKP formulation with 2-BMC inequalities	29.75	26.00	14.42
	PCKP formulation and facet-defining clique inequalities	26.00	26.00	0.00

ties for the PCKP is beneficial in certain instances.

6 Computational Results

In this section we apply our clique constraints to realistic examples and demonstrate their efficacy in reducing solution times when added at the root node and in a branch and cut framework. The first class of instances stems from routing optimization problems in internet protocol networks, where traffic demands must be routed unsplit along shortest paths through the network; see [3] for example. In these routing optimization problems, each individual link of the network defines a PCK subproblem. The items in the PCK problem correspond to the possible routing paths that traverse the link; see Figure 11(a). The weights and the values of the items are given by the demand volumes that would be routed along these paths and the corresponding routing cost, respectively. The capacity of the knapsack is simply the bandwidth (or data rate) of the network link. The precedence relations among the items are defined by the subpath consistency (also known as the Bellman property) that is implied



(a) Paths across link (v_3, v_4) in the network.

(b) Precedence relations among the paths.

Figure 11: Precedence relations in the routing optimization instances.

by the shortest path routing paradigm for the corresponding paths: If path P is chosen as the routing path between its terminals $s(P)$ and $t(P)$ and Q is a subpath of P, then Q must be chosen as the routing path between its terminals $s(Q)$ and $t(Q)$ as well. The resulting precedence relations are $x_Q \geq x_P$ for all paths P and Q with $Q \subseteq P$. Figure 11(b) illustrates these relations for a small example.

The second class of instances arose from open pit mine production scheduling problems; see [7, 8] for example. In these instances, the items correspond to blocks that form a discrete representation of the orebody. The weight of an item is given by the rock tonnage of the corresponding block, while its value is given by the net profit of mining and processing the block. The capacity of the knapsack is the total rock tonnage that can be mined within a given time frame. The precedence relations among the items correspond to the geological restrictions on the shape of the open pit. In order to remove a block, all blocks situated in an inverted cone above it must be removed first.

Table 6 shows the computational results obtained with CPLEX 11.2 using default settings and the best-bound branch and bound node selection strategy on a Linux PC with 4GB RAM and Intel Core2 Duo E8400 CPU with 3.00 GHz. In all cases, the LP-gap is reported at "node 0+" after all CPLEX preprocessing and automatic cutting plane generation has been applied at the root node. All CPU times reported include the time taken for any auxilliary calculations such as constructing conflict graphs and finding cliques. The columns labelled "Standard Formulation" in Table 6 describe the results of solving (PCKP) directly in CPLEX.

As is typical for realistic instances of the PCKP, the test instances do not satisfy Assumption 1. The columns labelled "with Fixing" in Table 6 demonstrate the improvements obtained by the simple, but very effective, fixing of variables $x_i = 0$ whenever $H(\{i\}) > b$. The solution times are dramatically reduced in all but one instance. The times are reduced by 93% in total and 70% on average for the telecommunication instances and by 24% in total and 17% on average for the mining instances. Note that CPLEX was not able to deduce all of these fixings automatically even with the most aggressive preprocessing and variable probing settings.

The columns labelled "with Cliques at Root" in Table 6 describe the results of adding clique inequalities at the root node in a cutting plane approach. In our algorithm, we simultaneously search for violated clique inequalities of both types (5) and (7). For this purpose, we build an extended conflict graph $CG' = (\mathcal{N} \cup \mathcal{N}', E \cup \mathcal{A}')$. This graph contains two nodes $i \in \mathcal{N}$ and $i' \in \mathcal{N}'$ for each item $i \in \mathcal{N}$. Node i corresponds to the variable x_i, while node i' corresponds its complement $1 - x_i$. CG' contains the edges $\{i, j\}$ for all $i, j \in \mathcal{N}$ with $H(\{i, j\}) > b$ and the edges $\{i, j'\}$ for all $i, j \in \mathcal{N}$ with $j \in A_i$. The first set of edges is exactly the edge set E of the (plain) conflict graph CG, the second set of edges forms the edge set $\mathcal{A}'$. It is easy to verify that any inclusion-wise maximal clique $\mathcal{C}'$ in CG' is either of the form $\mathcal{C}' = \mathcal{C} \subseteq \mathcal{N}$ or of the form $\mathcal{C}' = \mathcal{C} \cup \{i'\}$ with $\mathcal{C} \subseteq \mathcal{N}$ and $i' \in \mathcal{N}'$, where $\mathcal{C}$ is an inclusion-wise maximal clique in CG and, in the latter case, $i \in \mathcal{P}(\mathcal{C})$. Finding the most violated inequality of type (5) and (7) for a given fractional solution $\hat{x}$ is equivalent to finding a maximum weight clique in the extended conflict graph CG' with node weights $\hat{x}_i$ for the nodes $i \in \mathcal{N}$ and $1 - \hat{x}_i$ for the nodes $i' \in \mathcal{N}'$. In general, the problem of finding a maximum clique is NP-hard. However, there are numerous efficient heuristics and exact solution methods for finding maximum weight cliques in general graphs that can be applied. In our implementation we use the Sequential Greedy heuristic and the branch and bound algorithm proposed in [5] with a limit of at most 100 branch and bound nodes to explore. These algorithms proved to be efficient for the separation of clique inequalities in the general purpose integer programming solver SCIP [1]. Once we find a maximal clique $\mathcal{C}'$, if $\mathcal{C}' \subseteq \mathcal{N}$, we add the clique inequality (5) for $\mathcal{C} = \mathcal{C}'$. Otherwise, if $\mathcal{C}'$ is of the form $\mathcal{C}' = \mathcal{C} \cup \{i'\}$, we add the violated clique inequality (7). We terminate our cutting plane approach when we can no longer find a clique for which the corresponding clique inequality (5) or (7) cuts off the current fractional solution. The results in Table 6 show that the LP-gap decreases in all instances, and by a large amount in many instances.

Also the number of branch and bound nodes decreases significantly in all but one instance. In total, the reduction is roughly 50%. The solution times are mixed with gains offset by the necessary auxiliary clique calculations and the longer times needed to resolve the more difficult linear programs in the branch and bound tree. This increase, however, is only less than 3% in total.

The columns labelled “with Cliques in Tree” in Table 6 describe the results of applying the above cutting plane procedure within the branch and bound tree, using the conflict graph constructed at the root. The node weights in the conflict graph are updated to reflect the current fractional solution, but new conflicts that may arise deeper in the tree are not added. We see no further improvements in the number of branch and bound nodes and computation times compared to adding clique inequalities only at the root node. Apparently, the most beneficial cuts are found already at the root node. Adding cuts that are only violated in some parts of search tree seems to only marginally improve the lower bounds obtained at the corresponding subproblems, but to generate solutions which are more fractional, which, in consequence, leads to more branch and bound nodes to explore.

We also performed computations where the arcs of the conflict graphs were updated at the nodes of the branch and bound tree, incorporating also the variable fixings made by the branching decisions that lead to the current branch and bound node. This approach yields denser conflict graphs and more violated clique inequalities deeper in the branch and bound tree, but these inequalities are valid only locally for the current node and its descendents. The gains in the branch and bound process, however, were dominated by the expensive auxiliary calculations necessary to update the conflict graph at every node; therefore we do not report these results in detail.

7 Conclusions and Future Work

We have presented a new approach for determining facets of the PCKP polyhedron based on clique inequalities. The conditions derived in Section 3 can be checked to determine whether clique inequalities derived from the conflict graph are facet-defining whenever the problem instance contains pairs of items that cannot be included in the knapsack together. A procedure to generate a facet-defining clique inequality from any maximal clique in the conflict graph is also presented in Section 3. A comparison with previous polyhedral approaches to the PCKP based on knapsack cover-like inequalities in Sections 4 and 5 has demonstrated that the clique-based approach can generate facet-defining inequalities that cannot be found through the cover-based approach of previous authors. We provided a thorough classification of PCKP covers and cliques, and demonstrated the relationships between them. We showed in small examples that the addition of facet-defining clique-based inequalities for the PCKP is highly beneficial and that existing approaches are unable to reproduce our clique-based inequalities in some instances. We also conducted a numerical study of larger realistic PCK instances, adding our clique-based inequalities in a cutting plane approach at the root node and in a branch and cut framework. We found significant computational gains from adding the clique-based inequalities, especially in terms of reducing the LP relaxation gap and the number of branch and bound nodes required.

References

[1] Achterberg, T.: Constraint Integer Programming, PhD Thesis, Technische Universität Berlin (2007)

[2] Balas, E.: Disjunctive Programming. Ann. Discr. Math. **5**, 3-51 (1979)

[3] Bley, A.: Routing and Capacity Optimization for IP Networks. PhD Thesis, Technische Universität Berlin (2007)

[4] Bomze, I., Budinich, M., Pardalos, P. and Pelillo, M.: The maximum clique problem, In *Handbook of Combinatorial Optimization*, volume 4, Kluwer Academic Press (1999)

[5] Borndörfer, R. and Kormos, Z.: An Algorithm for Maximum Cliques, Unpublished working paper, Konrad-Zuse-Zentrum für Informationstechnik Berlin (1997)

[6] Boyd, E.A.: Polyhedral results for the precedence-constrained knapsack problem. Discrete Appl. Math. **41**, 185-201 (1993)

[7] Caccetta, L. and Hill, S.P.: An application of branch and cut to open pit mine scheduling, Journal of Global Optimization **27**, 349-365 (2003)

[8] Fricke, C.: Applications of Integer programming in Open Pit Mining, PhD Thesis, University of Melbourne (2006)

[9] Garey, M.R. and Johnson, D.S.: Computers and intractability: A guide to the theory of NP-completeness, W.H. Freeman and Company, San Francisco (1979)

[10] Ibarra, O.H. and Kim, C.E.: Approximation algorithms for certain scheduling problems. Math. Oper. Res. **4**, 197-204 (1978)

[11] Johnson, D.S. and Niemi, K.A.: On knapsacks, partitions, and a new dynamic programming technique for trees. Math. Oper. Res. **8**, 1-14 (1983)

[12] Nemhauser, G.L. and Wolsey, L.A.: Integer and Combinatorial Optimization, Wiley-Interscience, New York (1988)

[13] Park, K. and Park, S.: Lifting cover inequalities for the precedence-constrained knapsack problem. Discrete Appl. Math. **72**, 219-241 (1997)

[14] Shaw, D.X., Cho, G. and Chang, H.: A depth-first dynamic programming procedure for the extended tree knapsack problem in local access network design. Telecommunication Systems **7**, 29-43 (1997)

[15] Stecke, K.E. and Kim, I.: A study of part type selection approaches for short-term production planning. Int. J. Flexible Manufacturing Systems **1**, 7-29 (1988)

[16] Van de Leensel, R.L.M.J., van Hoesel, C.P.M. and van de Klundert, J.J.: Lifting valid inequalities for the precedence constrained knapsack problem. Math. Program. **86**, 161-185 (1999)

Appendix A: Proofs of Technical Results

Lemma 1(v): **Proof.** Let $i \in B \subseteq \mathcal{N}$ satisfy $A(B \setminus \{i\}) \subsetneq A(B)$. Let $j \in A(B) \setminus \{i\}$. Now $j \in A(B)$ so $A_j \subseteq A(B)$ by Lemma 1(i). Suppose $A_j \not\subseteq A(B) \setminus \{i\}$. Then $i \in A_j$, and hence $A_i \subseteq A_j$, by Property 1. Suppose further that $j \notin A(B \setminus \{i\})$. Then $j \in A_i$, and so $A_j \subseteq A_i$. Thus $A_i = A_j$ and so $i = j$, which contradicts the definition of j. Thus it must be that $j \in A(B \setminus \{i\})$, and so $A_j \subseteq A(B \setminus \{i\})$, by Lemma 1(i). But

$$A(B) = A(B \setminus \{i\}) \cup A_i \subseteq A(B \setminus \{i\}) \cup A_j = A(B \setminus \{i\})$$

since $A_i \subseteq A_j$ and $A_j \subseteq A(B \setminus \{i\})$. This contradicts the condition that $A(B \setminus \{i\}) \subsetneq A(B)$. Thus it must be that $A_j \subseteq A(B) \setminus \{i\}$. To conclude the proof that $A(B) \setminus \{i\}$ is a feasible packing, we note that since $H(B) \leq b$ and $A(B) \setminus \{i\} \subseteq A(B)$, it follows that $H(A(B) \setminus \{i\}) \leq b$. ■

Lemma 2 **Proof.** : Let $\mathcal{C} \subseteq \mathcal{N}$ be a clique in the conflict graph CG.

We first prove part (i). Let $i \in \mathcal{C}$. Suppose that there exists $j \in A_i \cap \mathcal{C}$, with $j \neq i$. By Property 1, $A_j \subseteq A_i$ so $A_i \cup A_j = A_i$. By the definition of the conflict graph, and since $\mathcal{C}$ is a clique, it must be that $H_i = H(\{i, j\}) > b$, which contradicts Assumption 1. Thus $(A_i \setminus \{i\}) \cap \mathcal{C} = \emptyset$. Obviously $i \in A_i$ and $i \in \mathcal{C}$, and the result follows.

Part (ii) is a simple consequence of part (i). We have that $A_i \subseteq A(\mathcal{C})$, so it follows that $A_i \setminus \{i\} \subseteq A(\mathcal{C})$. But $(A_i \setminus \{i\}) \cap \mathcal{C} = \emptyset$ by Lemma 2(i), and hence $A_i \setminus \{i\} \subseteq A(\mathcal{C}) \setminus \mathcal{C}$.

Finally we prove part (iii). Let $k \in A(\mathcal{C}) \setminus \mathcal{C}$. Then $k \in A(\mathcal{C})$, and there exists $i \in \mathcal{C}$ such that $k \in A_i$. It follows from Property 1 that $A_k \subseteq A_i$, and since $k \neq i$, we have that $A_k \subseteq A_i \setminus \{i\}$. Suppose that $A_k \not\subseteq A(\mathcal{C}) \setminus \mathcal{C}$. Then there must exist $j \neq i$ such that $j \in A_k \cap \mathcal{C}$. By Property 1 it follows that $A_j \subseteq A_k \subseteq A_i$, so $A_j \cup A_i = A_i$. But $i, j \in \mathcal{C}$, and thus by the definition of $\mathcal{C}$, $H(\{i, j\}) = H_i > b$. This contradicts Assumption 1. Hence $A_k \subseteq A(\mathcal{C}) \setminus \mathcal{C}$. ■

Lemma 3: **Proof.** $A_h \cup A_k$ is a feasible packing by Lemma 1(iii), and, since $k \notin A_h$, $(A_h \cup A_k) \setminus \{k\}$ is feasible packing by Lemma 1(v), (taking $B = \{h, k\}$ and $i = k$).

By Lemma 2(i), $A_h \cap \mathcal{C} = \{h\}$. Now by Assumption 1, the definition of the conflict graph CG, and since $H(\{h, k\}) \leq b$, it must be that $A_k \cap \mathcal{C} \subseteq \{h\}$. Hence $(A_h \cup A_k) \cap \mathcal{C} = (A_h \cap \mathcal{C}) \cup (A_k \cap \mathcal{C}) = \{h\}$. Finally, since $k \neq h$, ($h \in \mathcal{C}$ and $k \notin \mathcal{C}$), it follows that $((A_h \cup A_k) \setminus \{k\}) \cap \mathcal{C} = \{h\}$ also. ■

Instance			Standard Formulation			with Fixing				with Cliques at Root				with Cliques in Tree		
Name	$\|\mathcal{N}\|$	$\|\mathcal{S}\|$	B&B Nodes	LP-gap (%)	Time (sec)	Fixed	B&B Nodes	LP-gap (%)	Time (sec)	Clique cuts	B&B Nodes	LP-gap (%)	Time (sec)	Clique cuts	B&B Nodes	Time (sec)
Telecommunication Instances																
A	972	1661	153	9.89	0.62	632	330	8.79	0.23	40	88	4.40	0.49	91	80	0.65
B	981	1688	215	11.69	0.67	649	311	7.44	0.17	23	112	3.38	0.29	82	86	0.54
C	1336	2382	499	6.60	2.29	897	434	7.60	0.33	26	208	3.46	0.49	110	132	0.67
D	1790	3130	140	5.83	0.67	914	133	5.83	0.58	16	57	2.28	0.60	53	55	0.96
E	1790	3130	81	11.62	0.52	914	88	11.62	0.45	21	50	4.22	0.80	53	45	1.06
F	3091	5715	570	1.97	15.71	2455	225	1.89	0.27	14	196	1.38	0.45	133	220	1.14
G	3091	5715	23	13.90	0.70	3034	28	13.90	0.02	16	4	0.26	0.02	16	4	0.02
H	3091	5715	537	21.42	4.52	1712	328	23.06	0.99	4	163	12.74	0.98	53	119	2.02
I	3091	5715	1155	13.29	17.81	295	222	11.84	7.17	4	259	11.21	8.69	83	368	21.89
J	9235	17082	3253	17.91	184.31	6032	1101	17.91	6.08	14	196	14.31	5.93	139	136	17.46
K	9235	17082	681	22.45	74.66	6032	100	22.45	3.21	1	62	21.88	3.05	67	56	7.43
Mining Instances																
L	349	2101	113	32.17	0.38	46	127	28.68	0.19	3	92	8.72	0.18	5	91	0.20
M	538	3033	106	33.44	0.87	76	155	34.83	0.84	25	153	17.67	1.73	56	147	1.60
N	1217	7616	64	4.52	0.77	55	63	4.70	0.56	6	37	1.19	0.64	6	37	0.71
O	1711	11661	91	79.58	7.64	223	86	77.90	5.88	48	59	13.80	13.02	73	52	14.82
P	3243	22306	113	179.72	21.80	608	125	179.06	17.55	94	61	96.35	38.51	179	40	53.48
Q	3428	19555	815	45.67	42.34	488	1441	45.13	39.42	10	539	23.77	18.46	163	1451	66.48
R	4281	24452	2130	21.05	94.19	172	2178	20.93	74.91	69	974	11.90	59.56	193	1016	85.20
S	5624	36504	206	17.02	22.20	85	246	17.06	31.77	172	179	9.84	60.83	220	187	76.88
T	6271	42080	1078	87.25	481.93	773	993	87.86	206.13	50	775	47.64	161.70	337	865	315.40
U	6494	48626	126	100.87	91.98	352	100	100.19	91.76	218	104	79.34	248.58	381	134	345.56
V	10001	63944	1188	52.63	902.22	497	1226	52.52	714.79	292	578	44.68	574.60	1454	958	1601.40
W	11757	83218	1278	82.58	2504.92	650	1470	82.66	1987.41	366	1168	64.62	2079.78	1184	1347	3886.24

Table 6: Computational Results

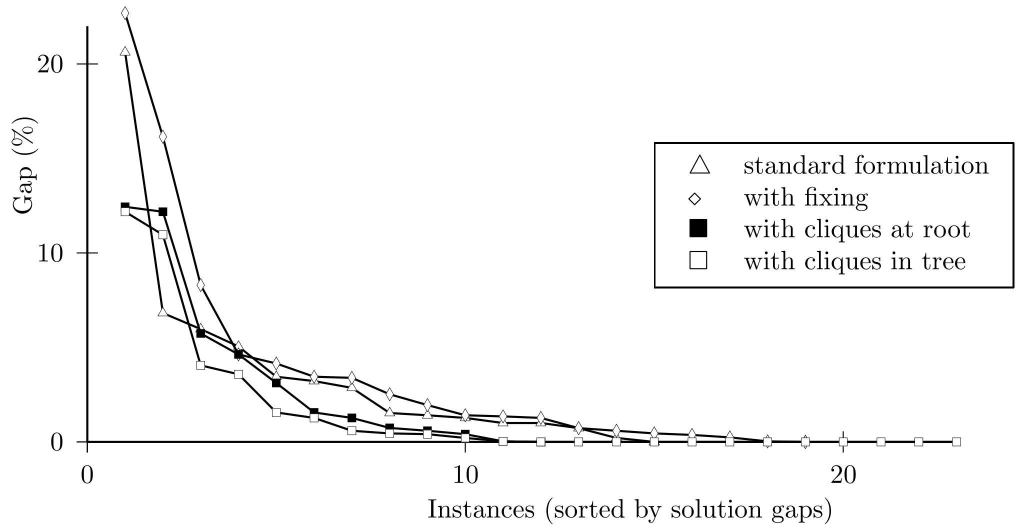


Table 7: Optimality gaps of solutions found after 50 branch-and-bound nodes.

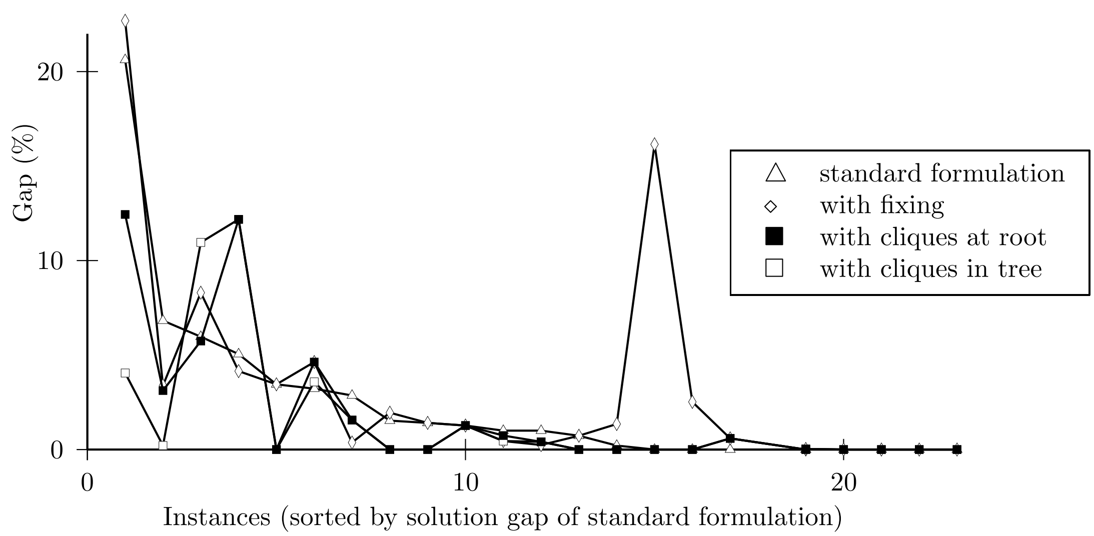


Table 8: Optimality gaps of solutions found after 50 branch-and-bound nodes.

Table 9: Summary of Notation

Notation	Definition
$\mathcal{N}$	the set of items available for inclusion in the knapsack.
$\mathcal{S}$	the set of all immediate precedence relationships in the problem instance.
$G = (\mathcal{N}, \mathcal{S})$	the directed graph representing the immediate precedence relationships in the problem instance.
$\mathcal{A}$	the set of all precedence relationships in the problem instance.
c_i	the value of item $i \in \mathcal{N}$, $c_i \in \mathbb{Z}$.
a_i	the weight of item $i \in \mathcal{N}$, $a_i \in \mathbb{Z}^+$.
b	the capacity of the knapsack, $b \in \mathbb{Z}^+$.
S_i	the set of immediate predecessors of item $i \in \mathcal{N}$.
A_i	the entire precedence set of item $i \in \mathcal{N}$ (including item i).
B	a set of items, $B \subseteq \mathcal{N}$.
$A(B)$	the union of the entire precedence sets for the items in the set B, $A(B) = \cup_{i \in B} A_i$.
H_i	the capacity required for item i to be included in the knapsack, $H_i = \sum_{j \in A_i} a_j$.
$H(B)$	the total capacity required to include all items in the set B, $H(B) = \sum_{j \in A(B)} a_j$.
D_i	the entire successor set of item i (including item i).
e_i	the i^{th} standard basis vector in $\mathbb{R}^{\|\mathcal{N}\|}$.
$x(B)$	the characteristic vector of the set B, $x(B) = \sum_{i \in B} e_i$.
$\hat{J}_B(k)$	the descendent set of k in the set B, $\hat{J}_B(k) = \{j \in B : k \in A_j\}$ for each $k \in A(B) \setminus B$.
P	the PCKP feasible set defined by (2)-(4).
$conv(P)$	the convex hull of the PCKP feasible set P.
$CG = (\mathcal{N}, E)$	a conflict graph with edge $\{i, j\} \in E$ if and only if $H(\{i, j\}) > b$.
E	the set of edges in the conflict graph CG.
$\mathcal{C}$	a set of items that is a clique in the conflict graph CG, $\mathcal{C} \subseteq \mathcal{N}$.
$\mathcal{P}(\mathcal{C})$	the set of all items in the intersection of the entire precedence sets of all the items in the clique $\mathcal{C}$, $\mathcal{P}(\mathcal{C}) = \cap_{j \in \mathcal{C}} A_j$.
$\mathcal{Q}(\mathcal{C})$	the set of all items in the intersection of the entire precedence sets of all the items in the clique $\mathcal{C}$, with no items in their entire successor sets D_i that satisfy the same property, $\mathcal{Q}(\mathcal{C}) = \{i \in \mathcal{P}(\mathcal{C}) : \mathcal{C} \nsubseteq D_k \text{ for all } k \in D_i \setminus \{i\}\}$.
C	a set of items that is a cover for an instance of the PCKP, $C \subseteq \mathcal{N}$.
(K-)BMC	a (K-)Boyd minimal cover.
(K-)MIC	a (K-)minimal induced cover.
$P(B)$	the convex hull of feasible solutions to (PCKP) restricted to those variables in $A(B)$, $P(B) = conv(proj_{A(B)}\{x(D) \in P : D \subseteq A(B)\})$.