Appendix

Target Cuts from Relaxed Decision Diagrams

Christian Tjandraatmadja, Willem-Jan van Hoeve
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, ctjandra@andrew.cmu.edu, vanhoeve@andrew.cmu.edu

---

Appendix A: Proof of Corollary 1

Corollary 1. Let B be a BDD representing S. The set of binary points of any face of $\mathrm{conv}(S)$ is representable by a sub-BDD of B.

Proof. Let F be a face of $\mathrm{conv}(S)$. Let ω be a point in the relative interior of $\mathrm{conv}(S)$. Denote by S_ω the translation of S so that it contains the origin in its relative interior, that is, $S_\omega = S - \omega$. Additionally, consider the similarly translated $F_\omega = F - \omega$.

Consider a valid inequality $u^\top x \leq v$ defining F_ω. That is, $\{x \in \mathrm{conv}(S_\omega) : u^\top x = v\} = F_\omega$. Since the origin is satisfied by the inequality, $v \geq 0$. Without loss of generality, we consider the cases $v = 0$ and $v = 1$, since the inequality can be normalized when $v > 0$.

Suppose the inequality is of the form $u^\top x \leq 0$. Then it must be an implicit equality – in other words, $u^\top x = 0$ is valid – because otherwise the origin would not be in the relative interior of $\mathrm{conv}(S_\omega)$. Then $F_\omega = S_\omega$, and thus $F = S$, which is representable by B.

Suppose the inequality is of the form $u^\top x \leq 1$. This implies $u \in S_\omega^\circ$. By Theorem 3, there exists v such that $(u, v) \in P_\omega^*(B)$. We call an arc tight if its corresponding constraint for $P_\omega^*(B)$ is tight for (u, v). That is, an ℓ-arc (i, j) at layer k is tight if $v_j = v_i - \ell u_k$ (or, if (i, j) is a long arc, $v_j = v_i - \sum_p \ell_p u_p$). We call a path tight if it is composed only of tight arcs. Let T be the set of arcs that are in at least one tight path from the root s to the terminal t. The subgraph of B induced by T forms a sub-BDD B'.

Let S' be the set of points represented by B'. Then $x \in S'$ if and only if x corresponds to a tight path from s to t, or equivalently $u^\top x = 1 + u^\top \omega$ by summing up all constraints of the path. This implies $S' = \{x \in \mathrm{conv}(S) : u^\top (x - \omega) = 1\} = F_\omega + \omega = F$. Therefore B' represents the set of binary points of the face F.

□

Appendix B: Algorithm to Compute the Center of Points Represented by a BDD

We present an alternative method to compute the center of S that is more suitable to our context, relative to the one proposed by Behle (2007). This may be useful to obtain an interior point for the target cut approach.

The first step is to compute, for each arc (i,j), the number of paths n_{ij} from root s to terminal t that traverse (i,j). To do so, observe that $n_{ij} = n_i^- + n_j^+$, where n_i^- is the number of paths from s to i and n_j^+ is the number of paths from j to t. Calculating the former can be done in a single top-down pass: at node s we have $n_s^- = 1$, and at each node $i \neq s$, we let $n_i^- = \sum_{j \in N^-(i)} n_j^-$, where $N^-(i)$ denotes the parents of i. Likewise, the latter can be computed in a bottom-up pass: $n_t^+ = 1$ and $n_i^+ = \sum_{j \in N^+(i)} n_j^+$, where $N^+(i)$ denotes the children of i.

We then use n_{ij} to calculate the total number $N_{\ell,k}$ of paths with label ℓ at layer k. That is, $N_{\ell,k} := \sum_{(i,j) \in S_{\ell,k}} n_{ij}$, where $S_{\ell,k}$ is the set of ℓ-arcs in layer k. Since paths of B correspond to points of S, $N_{\ell,k}$ is the total number of points $x \in S$ such that $x_k = \ell$. In addition, note that $N := \sum_{(i,j) \in \delta^+(s)} n_{ij}$ is the total number of points of S. Therefore, the center of S can be expressed as $\frac{1}{N}(0N_{0,1} + 1N_{1,1}, \ldots, 0N_{0,n} + 1N_{1,n}) = \frac{1}{N}(N_{1,1}, \ldots, N_{1,n})$.

Appendix C: Proof of Theorem 5 on Face Certificates

We first restate Theorem 5, which shows how to obtain a face certificate for a target cut from a decision diagram. All definitions required for the theorem, such as (P), (D), $\tilde{P}_{\text{flow}}$, and $S^+_{\alpha^*}$, can be found in Section 6.

THEOREM 5. Let (u^*, v^*) and f^* be an optimal primal-dual pair for (P) and (D). If f^* is an extreme point of $\tilde{P}_{\text{flow}}$ and α^* is the result of a standard flow decomposition applied to f^*, then $S^+_{\alpha^*}$ is a $|S^+_{\alpha^*}|$-face certificate for $u^{*\top} x \leq 1$ with respect to $\text{conv}(S)$.

We prove a series of intermediate results and the theorem will follow. For simplicity, throughout this section we assume the origin is in the interior of $\text{conv}(S)$. At the end of this section, we argue that all of the following results still hold without this assumption.

We put aside decision diagrams for a moment and briefly discuss how to obtain certificates if we had the set of points S explicitly. In this scenario, target cuts may be generated by solving $\max_u \{u^\top \bar{x} : u \in S^\circ\}$, denoted by ($\tilde{\text{P}}$). Its dual is $\min_\alpha \{\sum_{x \in S} \alpha_x : \alpha \in P_{\text{cone}}\}$, denoted by ($\tilde{\text{D}}$), where

$$P_{\text{cone}} = \{\alpha : \tilde{S}\alpha = \bar{x},\ \alpha \geq 0\}$$

and $\tilde{S}$ is the matrix formed by the points of S in its columns. In other words, P_{cone} is the polyhedron of the coefficients of the conic combinations that express $\bar{x}$.

In the following proposition, we view a basis J of $\tilde{S}$ as a set of points of S, since $\tilde{S}$ represents a set of points of S in its columns. Moreover, we call a $\dim(S)$-face certificate a *facet certificate*.

PROPOSITION 2. *Let u^* and α^* be an optimal primal-dual pair for ($\tilde{P}$) and ($\tilde{D}$).*

(i) If J is an optimal basis for α^*, then J is a facet certificate for $u^{*\top} x \leq 1$ with respect to conv(S).

(ii) If α^* is an extreme point of P_{cone}, then $S^+_{\alpha^*}$ is a $|S^+_{\alpha^*}|$-face certificate for $u^{*\top} x \leq 1$ with respect to conv(S).

Proof. Since basic variables have zero reduced costs, $x \in J$ implies $u^{*\top} x = 1$. Therefore, all points in J are tight with respect to $u^{*\top} x \leq 1$. Moreover, J is a maximal linearly independent set by definition, and thus J is a facet certificate.

If we do not have an optimal basis but α^* is an extreme point of P_{cone}, then there exists a basis associated to α^* that contains $S^+_{\alpha^*}$. Therefore, all points in $S^+_{\alpha^*}$ are tight with respect to $u^{*\top} x \leq 1$ and are linearly independent, implying $S^+_{\alpha^*}$ is a $|S^+_{\alpha^*}|$-face certificate.

□

The weaker result (ii) from Proposition 2 is useful to prove Theorem 5. In the decision diagram case, we cannot obtain a facet certificate from (P) in general without modifying its optimal u^* because u^* is not guaranteed to define a facet without additional steps, as discussed in Section 5.4.

Our goal now is to derive a similar result for the case where S is implicitly represented as a decision diagram. That is, instead of ($\tilde{\text{P}}$), we solve (P). Recall that the dual (D) of (P) is $\min_f \{\sum_{j \in \delta^+(s)} f_{sj} : f \in \tilde{P}_{\text{flow}}\}$. In this section, it is convenient to express the constraints of $\tilde{P}_{\text{flow}}$ using matrix notation:

$$\tilde{P}_{\text{flow}} = \{f : Nf = 0,\ Vf = \bar{x},\ f \geq 0\}.$$

Here, N is the node-arc incidence matrix and V is the arc value matrix. That is, N_{ie} is 1 if arc e incides on vertex i and 0 otherwise and V_{ke} is the value that arc e assigns to variable x_k.

In order to link $\tilde{P}_{\text{flow}}$ to P_{cone}, we rely on the following polyhedron that maps a fixed $f \in \tilde{P}_{\text{flow}}$ to $\alpha \in P_{\text{cone}}$:

$$P_{\text{dec}}(f) = \{\alpha : H\alpha = f, \alpha \geq 0\},$$

where H is the arc-path incidence matrix, that is, H_{ep} is 1 if arc e is in path p or 0 otherwise. This polyhedron represents all possible ways that a fixed $f \in \tilde{P}_{\text{flow}}$ can be decomposed into $\alpha \in P_{\text{cone}}$, which is shown in the next proposition.

Proposition 3. *If $\alpha \in P_{cone}$, then $H\alpha \in \tilde{P}_{flow}$. Conversely, if $f \in \tilde{P}_{flow}$ and $\alpha \in P_{dec}(f)$, then $\alpha \in P_{cone}$.*

Proof. First note that all entries in the NH matrix are zero: each node has a single incoming arc and a single outgoing arc in a path, which cancel out when calculating NH. In addition, $VH = \tilde{S}$, since $(VH)_{kp}$ corresponds to the value assigned to x_k by the path p.

Let $\alpha \in P_{\text{cone}}$. Then $N(H\alpha) = 0$, $V(H\alpha) = \tilde{S}\alpha = \bar{x}$, and $H\alpha \geq 0$. Therefore, $H\alpha \in \tilde{P}_{\text{flow}}$.

Let $f \in \tilde{P}_{\text{flow}}$ and $\alpha \in P_{\text{dec}}(f)$. Then $\tilde{S}\alpha = VH\alpha = Vf = \bar{x}$ and $\alpha \geq 0$. Therefore, $\alpha \in P_{\text{cone}}$.

□

Next, we prove two lemmas that together imply Theorem 5.

Lemma 1. *Let α be the result of a standard flow decomposition applied to $f \in \tilde{P}_{flow}$. Then α is an extreme point of $P_{dec}(f)$.*

Proof. It suffices to show that the columns x of $\tilde{S}$ such that $\alpha_x > 0$ are linearly independent. The reason is that, if so, there must exist a basis J of $\tilde{S}$ containing these columns. Moreover, J must be associated to α since $\tilde{S}_J \alpha_J = \tilde{S}\alpha = \bar{x}$, given that $\alpha_x = 0$ for all $x \notin J$. This implies α is an extreme point.

Let t be the number of iterations of the decomposition, or equivalently the number of points x with $\alpha_x > 0$ at the end of the decomposition. Consider a $t \times t$ matrix M where the columns (paths) are the ones with positive flow from the decomposition in the order they were encountered, and the rows (arcs) are the bottleneck arcs of each of those paths also in the same order. Then M is lower triangular since a bottleneck arc never reappears in a path after its flow is set to zero. Therefore, these columns are linearly independent.

□

The following result connects extreme points of $\tilde{P}_{\text{flow}}$ and $P_{\text{dec}}(f)$ with extreme points of P_{cone}.

LEMMA 2. *If f is an extreme point of $\tilde{P}_{flow}$ and α is an extreme point of $P_{dec}(f)$, then α is an extreme point of P_{cone}.*

Proof. Suppose for contradiction there exist $\alpha^1, \alpha^2 \in P_\alpha$ such that $\alpha = \lambda\alpha^1 + (1-\lambda)\alpha^2$ with $\lambda > 0$. We show that if α^1 and α^2 correspond to different flows, then f is not an extreme point of $\tilde{P}_{\text{flow}}$, and if they correspond to the same flow, then α is not an extreme point of $P_{\text{dec}}(f)$.

Case 1: $H\alpha^1 \neq H\alpha^2$. Define $f^t = H\alpha^t$ for $t = 1, 2$. By Proposition 3, $f^t \in \tilde{P}_{\text{flow}}$. Moreover, $f = H\alpha = H(\lambda\alpha^1 + (1-\lambda)\alpha^2) = \lambda H\alpha^1 + (1-\lambda)H\alpha^2 = \lambda f^1 + (1-\lambda)f^2$, with f^1 and f^2 distinct. Thus f is not an extreme point of $\tilde{P}_{\text{flow}}$, a contradiction.

Case 2: $H\alpha^1 = H\alpha^2$. It suffices to show that $\alpha^1, \alpha^2 \in P_{\text{dec}}(f)$, thus proving that α is not an extreme point of $P_{\text{dec}}(f)$. We have $f = H\alpha = H(\lambda\alpha^1 + (1-\lambda)\alpha^2) = \lambda(H\alpha^1 - H\alpha^2) + H\alpha^2 = H\alpha^2$. This also implies $f = H\alpha^1$. Therefore, $\alpha^1, \alpha^2 \in P_{\text{dec}}(f)$.

□

Finally, we complete the proof of Theorem 5 by putting together the previous results.

Proof of Theorem 5. By Lemmas 1 and 2, α^* is an extreme point of P_{cone}. The result then follows from Proposition 2.

□

As discussed in Section 5.4, the cut generation algorithm may involve translating points if the origin is not in the interior of conv(S). While this may take away the linear independence of the points of S^+_α, they remain affinely independent, and thus all previous results still hold.

Appendix D: Decision Diagram Construction for Linear Inequalities

For the set covering problem, we use a generic BDD construction for linear inequalities. We provide a brief description of this construction in this section.

Given a set of m constraints $a_i^\top x \leq b_i$, each state carries m values representing the right-hand sides of each constraint after assigning a partial solution. More precisely, a state is a vector $s \in \mathbb{R}^m$ where $s_i = b_i - \sum_{j \in S_p} a_{ij}x_j$ and S_p is the set of variable indices that are already assigned at the state. Therefore, the transition function subtracts a_{ij} from s_i if x_j is assigned to one, and s_i is kept unchanged if x_j is assigned to zero. Merging non-equivalent states into a relaxed one is done by keeping the most relaxed right-hand side for each constraint.

In addition, we maintain the range that the left-hand side can take for each constraint in a state. This allows us to perform a number of simple checks in order to remove or merge nodes more often. More specifically, they are used to verify if a constraint is infeasible or is feasible for all remaining assignments. They are also used in a simple propagation scheme across constraints within a state.

With these additional checks, this construction applied to the set covering formulation turns out to be equivalent to the one specific to the set covering problem described by Bergman et al. (2016, Section 3.6), except that its genericness adds an overhead of time and memory.

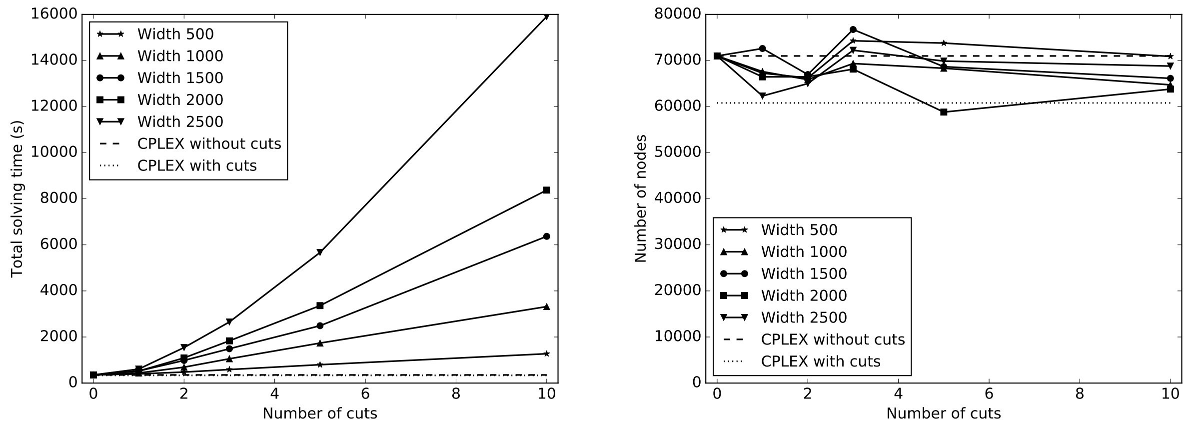

Figure 12 **Solving time (left) and number of nodes of branch-and-bound tree (right) for cuts on independent set instances of density 50% with 250 vertices.**

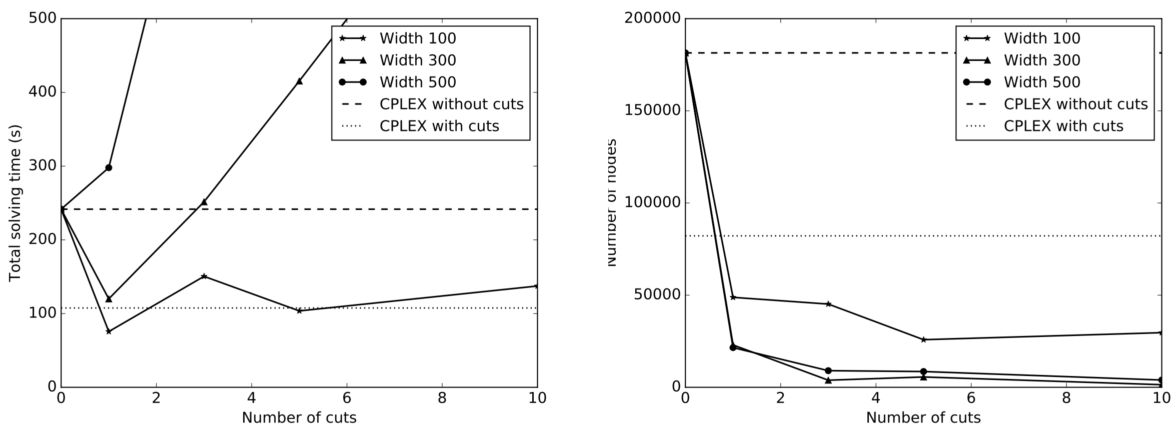

Figure 13 **Solving time (left) and number of nodes of branch-and-bound tree (right) for cuts on set covering instances of bandwidth 40.**

Appendix E: Additional Computational Results

In this section, we present additional graphs on computational results. Figure 12 depicts solving time and number of nodes for independent set instances of density 50%, in which we see target cuts have little effect due to the relatively weak BDD relaxation. Figure 13 shows solving time and number of nodes for set covering instances of bandwidth 40, which are similar to the results for bandwidth 50.

Figure 14 gives us a breakdown of where the time is spent for the first cut for independent set instances. In both situations, the time to construct a decision diagram is very small compared to the time to solve the LP to generate a cut – less than 2%. For density 80%, we can observe that it is not too time consuming to generate a cut and it significantly reduces the time spent in the branch-and-bound tree. This however does not happen for density 50%, where generating a cut is significantly more expensive than solving the root LP relaxation. We also see in density 80% how increasing the width of the BDD yields a stronger cut that significantly reduces the time spent in the branch-and-bound tree.

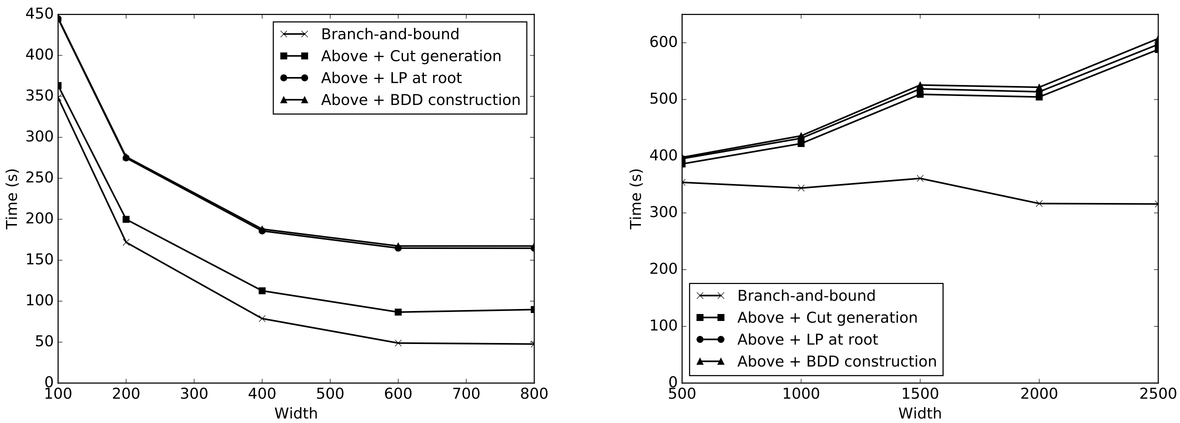

Figure 14 **Total solving time breakdown with a single cut for independent set instances of density 80% with 400 vertices (left) and 50% with 250 vertices (right).**

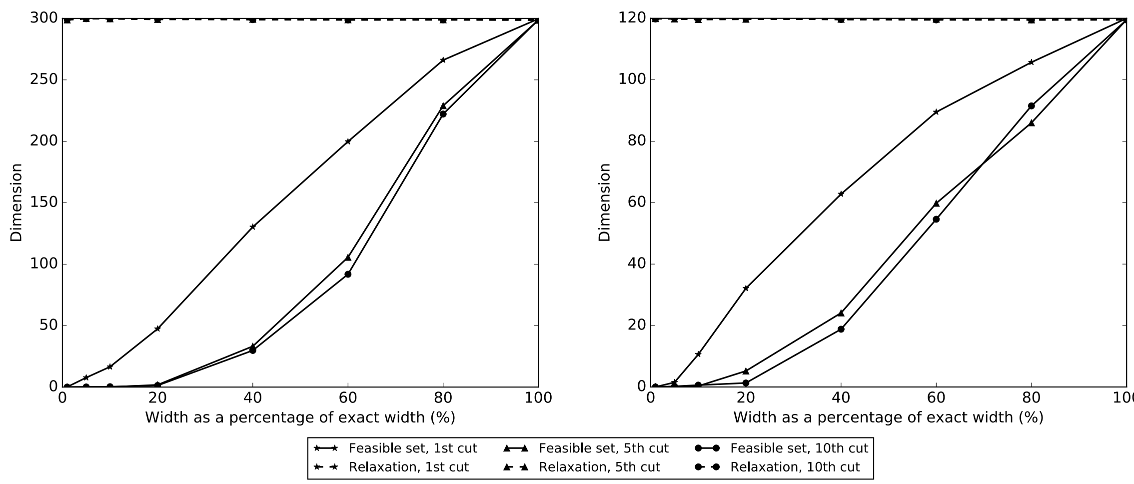

Figure 15 **Face dimensions for independent set instances of density 80% and 300 vertices (left) and density 50% and 120 vertices (right), after applying the perturbation heuristic to increase dimension.**

Figure 15 shows the same graphs from Figure 5, in which we examined the dimension of the faces defined by the cuts, except that in this case we apply the perturbation heuristic. If we compare them to Figure 5, we observe that the perturbation heuristic does as expected: it fully or almost fully increases the dimensions of the cuts with respect to the relaxation.

References

Behle M (2007) *Binary decision diagrams and integer programming*. Ph.D. thesis, Max Planck Institute for Computer Science, Saarbrücken, Germany.

Bergman D, Cire AA, van Hoeve WJ, Hooker JN (2016) *Decision Diagrams for Optimization* (Springer).