Generating Cuts from the Ramping Polytope for the Unit Commitment Problem

Ben Knueven and Jim Ostrowski
Department of Industrial and Systems Engineering University of Tennessee, Knoxville, TN 37996
bknueven@vols.utk.edu jostrows@utk.edu

Jianhui Wang
Energy Systems Division Argonne National Laboratory, Argonne, IL 60439
jianhui.wang@anl.gov

April 5, 2016

Abstract

We present a perfect formulation for a single generator in the unit commitment problem, inspired by the dynamic programming approach taken by Frangioni and Gentile. This generator can have characteristics such as ramping constraints, time-dependent start-up costs, and start-up/shut-down ramping. To develop this perfect formulation we extend the result of Balas on unions of polyhedra to present a framework allowing for flexible combinations of polyhedra using indicator variables. We use this perfect formulation to create a cut-generating linear program, similar in spirit to lift-and-project cuts, and demonstrate computational efficacy of these cuts in a utility-scale unit commitment problem.
1 Introduction

The unit commitment problem (UC) is that of scheduling generators to meet power demand, and has been one of the great successes of mixed-integer programming models. The Midwest Independent Transmission System Operator (MISO), recipient of the Edelman Award in 2011, reports annual savings of over $500 million by using MIP models over Lagrangian relaxation techniques for unit commitment [8]. Because of the scales involved, a 1% savings in energy markets is greater than $10 billion annually [22].

The unit commitment problem is nearly decomposable as generators are only linked through the demand constraint. Therefore most improvements in unit commitment models are a result of studying the properties of an individual generator’s production schedule. Frangioni and Gentile [13] provide a dynamic programming model for an individual generator with ramping constraints. Inspired by their dynamic programming model, we construct a compact extended formulation for a single generator, which can be used to model generators within a unit commitment MIP model. The formulation we develop can be used to model any properties of a generator that are polyhedrally representable when the commitment status is fixed. We then use this extended formulation to create a cut-generating linear program that can be used as a callback in a unit commitment mixed-integer programming model.

The rest of the paper is outlined as follows. Section 2 reviews the current state of the unit commitment problem, including the typical 3-binary formulation used in state-of-the-art models, and constructs the extended formulation which is the main result of this paper. Section 3 extends the classic result of Balas [3, 4] to prove that our extended formulation is valid, and does so in more general terms than we strictly require. In Section 4 we use the results of the preceding sections to develop a cut-generating linear program for a ramping-constrained generator, and we present some computational results based on a utility-scale unit commitment problem. Finally, in Section 5 we draw conclusions and discuss possible directions for future research.
2 The Unit Commitment Problem

We begin by providing an overview of the unit commitment problem. For a set of generators $G$ and $T$ time steps, we formulate the unit commitment problem as follows:

$$\min \sum_{g \in G} c^g(p^g) \quad (1a)$$

s.t. $\sum_{g \in G} p^g_t \geq D_t, \forall t \in [T] \quad (1b)$

$$\sum_{g \in G} \bar{p}^g_t \geq D_t + R_t, \forall t \in [T] \quad (1c)$$

$p^g$ is the power output vector of generator $g$, $c^g(p^g)$ is the cost of the vector $p^g$, and $\bar{p}^g_t$ is the maximum power available from generator $g$ at time $t$. $D_t$ is the electricity demand at time $t$, while $R_t$ is the spinning reserve requirement at time $t$. For convenience let $[T] := \{1, \ldots, T\}$. $\Pi^g$ represents the technical constraints on production and commitment of generator $g$, such as minimum up/down times, ramping rates, time-dependent start-up costs, etc. As mentioned above, most research in improving unit commitment models has focused improving the modeling of individual generators, i.e., (1d) above.

Most of the literature modeling individual generators followed Garver’s [14] general structure, using three different types of binary variables to describe the status of a generator at a given time: one variable indicating if the generator is on, another indicating if is turned on during the period, and the last indicating if generator is turned off during time period. Models of this type are referred to as 3-binary models, or 3-bin models. An alternative 1-binary model [9], or 1-bin model, considers the variables indicating if the generator is turned on/off as superfluous, rewriting each constraint using only variables that represent if the generator is on at a given time period. The hope in this reduction is that fewer binary variables will lead to smaller branch-and-bound trees and smaller computation times. However, moving to a smaller formulation comes with a cost of weaker inequalities. A convex hull description for a simplified generator using the 1-bin formulation is given in [19], showing that the convex hull has exponentially many constraints. Yet, the same simplified generator’s production region has a linearly-sized convex hull when using the 3-bin model [26].

The simplified generator considered in [26] only models on/off status, minimum/maximum power, and minimum
up/down times. For this reason, several recent results have strengthened the 3-bin model with additional generator characteristics. A common extension is to the case when ramping constraints are considered. Ramping constraints represent the fact that generators, in general, cannot vary their power output dramatically from one time period to the next. Polynomial classes of strengthening inequalities for the 3-bin model with ramping are given in [23]; [11] builds on this by providing exponential classes of such strengthening inequalities along with a polynomial separation algorithm. Additionally, [11] provides a convex hull description for the 3-bin model with ramping when only two time periods are considered; [25] extends this to three time periods.

A convex hull description for the 3-bin model with the addition of start-up and shut-down power is proved in [20]. This allows for the modeling of a generator’s output below economic minimum when starting up and shutting down. Their model also separates power from energy (often assumed to be the same since almost all UC models operate on a one-hour time interval) while still maintaining integrality.

First, we review the standard 3-bin model typically used to represent \( \Pi^g \) and \( c^g(\cdot) \). Then we turn our focus toward the dynamic programming method for optimizing over a single generator laid out in [13]. Indeed using a dynamic programming procedure as in [13] allows one to optimize a convex function over the unit commitment polytope (with ramping constraints) in polynomial time. It should not be surprising then to find polynomial-sized extended formulations for \( \Pi^g \) (although this is by no means guaranteed, see [28]). Additionally, the formulation derived can be trivially extended to model other generator characteristics, provided the constraints on the generator are represented with a polytope when the commitment status is fixed.

### 2.1 3-bin Formulation

We now describe the typical 3-bin formulation for the feasible region \( \Pi^g \) with cost function \( c^g(\cdot) \). Consider the binary vectors \( u^g, v^g, w^g \in \{0, 1\}^T \), where \( u^g_t \) is the commitment status of the generator at time \( t \), \( v^g_t \) indicates if the generator was started up at time \( t \), and \( w^g_t \) indicates if the generator was shut down at time \( t \). Suppose \( UT \) and \( DT \) are the minimum up and down time for the generator. We first consider the logical constraints [14]

\[
    u^g_t - u^g_{t-1} = v^g_t - w^g_t, \quad \forall t \in [T],
\]

(2)
and the minimum up/downtime constraints [26]

\[ \sum_{i=t-UT+1}^{t} v_i^g \leq u_i^g, \forall t \in [UT^g, T] \]  

(3)

\[ \sum_{i=t-DT+1}^{t} w_i^g \leq 1 - u_i^g, \forall t \in [DT^g, T]. \]  

(4)

Rajan and Takriti [26] showed that (2-4) along with the variable bound constraints give a convex hull description for the minimum up/downtime polytope.

Next we consider constraints on the generation limits. Let \( \bar{P} \) and \( P \) represent the minimum and maximum feasible power output, \( RD \) and \( RU \) represent the minimum and maximum ramping rates, and \( SD \) and \( SU \) represent the maximum shut-down and start-up levels. First we note that when a generator is on it must be operating within its specified limits

\[ P^g u_i^g \leq p_i^g \leq \bar{P}^g u_i^g, \forall t \in [T]. \]  

(5)

We note if \( RU^g, RD^g \geq (\bar{P}^g - P^g) \) and \( SU^g, SD^g \geq \bar{P}^g \), then (2 - 5) are a perfect formulation for this simple generator, as mentioned in Section 2. However, most generators are not so simple, and have ramp-up constraints:

\[ \bar{p}_{i}^g - p_{i-1}^g \leq RU^g u_{i-1}^g + SU^g v_i^g, \forall t \in [T], \]  

(6)

and ramp-down constraints:

\[ p_{i-1}^g - \bar{p}_i^g \leq RD^g u_i^g + SD^g w_i^g, \forall t \in [T]. \]  

(7)

For reference later, define:

\[ R^g \Pi_{3\text{-bin}} := \{(p^g, \bar{p}^g, u^g, v^g, w^g) \in \mathbb{R}^{5T} | (2 - 7); (u^g, v^g, w^g) \in [0, 1]^{3T}\} \]  

(8)

and

\[ \Pi^g_{3\text{-bin}} := \{(p^g, \bar{p}^g, u^g, v^g, w^g) \in R^g \Pi_{3\text{-bin}} | (u^g, v^g, w^g) \in \{0, 1\}^{3T}\}. \]  

(9)

That is, \( R^g \Pi_{3\text{-bin}} \) is the continuous relaxation of the technical constraints and \( \Pi^g_{3\text{-bin}} \) is the feasible set for the technical constraints for generator \( g \) in the 3-bin formulation. We will colloquially refer to \( R^g \Pi_{3\text{-bin}} \) as “3-bin space”, dropping the \( g \) when it is implied by context.
Now we consider the cost function $c^g(\cdot)$. Typically $c^g(p^g) = c^g_f(p^g) + \sum_{t \in [T]} c^g_p(p^g_t)$, where $c^g_f(\cdot)$ is convex and either quadratic or piecewise linear in the power output, and $c^g_f(\cdot)$ is the fixed commitment costs and start-up/shutdown costs, and as such is a function of the indicator variables. First, we consider $c^g_p(\cdot)$. We assume that $c^g_p(\cdot)$ is convex and piecewise linear where $1^g P^g, \ldots, L^g P^g$ represent the maximum power available for prices $1^g c^g < \ldots < L^g c^g$. Define $0^g P^g = 0$. We use the standard convex piecewise formulation then by introducing new variables $l^g p^g_t$, representing the power generator $g$ produces at time $t$ at cost $l^g c^g$, along with the constraints

$$0 \leq l^g p^g_t \leq l^g P^g - l^g - 1^g P^g, \forall t \in [L], \forall t \in [T] \quad (10a)$$

$$p^g_t = \sum_{l=1}^{L} l^g p^g_t, \forall t \in [T]. \quad (10b)$$

We can then represent $c^g_p(\cdot)$ linearly as $\sum_{t \in [T]} \sum_{l \in [L]} l^g c^g l^g p^g_t$. Now consider $c^g_f(\cdot)$. Typically the start-up cost is an increasing function of how long the generator has been off. For simplicity we will only consider two start-up types, hot (H) and cold (C). A start-up is said to be hot if the generator has been off for $\hat{t} \in [DT^g_H, DT^g_C]$ time periods, where $DT^g_H = DT^g$. We formulate the start-up costs as in Morales-España et al. [21]; namely, let $H^g \delta_t, C^g \delta_t \in \{0, 1\}$ represent a hot and cold start-up, respectively. Then we may write $H^g \delta^g_t, C^g \delta^g_t$ in terms of the start-up and shut-down variables $v^g_t, w^g_t$:

$$H^g \delta^g_t \leq \sum_{i=DT^g_H}^{DT^g_C - 1} w^g_{t-i}, \forall t \in [DT^g_C, T] \quad (11a)$$

$$H^g \delta^g_t + C^g \delta^g_t = v^g_t, \forall t \in [T]. \quad (11b)$$

Thus, if $R^g c^g$ is the fixed cost of running the generator, and $D^g c^g$ is the cost of shutting down the generator, then we can represent $c^g_f(\cdot)$ linearly as $\sum_{t \in [T]} (C^g c^g C^g \delta^g_t + H^g c^g H^g \delta^g_t + R^g c^g v^g_t + D^g c^g w^g_t)$.

If the ramping constraints (6) and (7) are binding, then it is well known that $\text{conv}(\Pi^g_{3\text{-bin}}) \neq R^g \Pi^g_{3\text{-bin}}$ (where $\text{conv}(S)$ is the convex hull of the set $S$). Recently, [11] characterized $\text{conv}(\Pi^g_{3\text{-bin}})$ for when $T = 2$ and [25] characterized $\text{conv}(\Pi^g_{3\text{-bin}})$ for $T = 3$. In the next section we will develop a new extended formulation for $\Pi^g$, which can be used to generate valid inequalities for $\text{conv}(\Pi^g_{3\text{-bin}})$. 6
2.2 The Feasible Dispatch Polytope

The feasible dispatch polytope describes the possible generator outputs given that the generator’s on/off status has been fixed. Let $D^{[a,b]} \subset \mathbb{R}^T$ represent the set of all feasible production schedules assuming that the generator is only (and continuously) on during the time interval $[a, b]$. For any $p^{[a,b]} \in \mathbb{R}_+^T$ in $D^{[a,b]}$, $p_i^{[a,b]}$ represents the power produced by the generator at time $i$ (note that $p_i^{[a,b]} = 0$ for all $i$ not in the interval $[a, b]$), and $\bar{p}_i^{[a,b]}$ represents the maximum power available at time $i$. We have that

$$D^{[a,b]} = \{p^{[a,b]}, \bar{p}^{[a,b]} \in \mathbb{R}_+^T | A^{[a,b]} p^{[a,b]} + \bar{A}^{[a,b]} \bar{p}^{[a,b]} \leq b^{[a,b]} \} \quad (12)$$

for $A^{[a,b]}, \bar{A}^{[a,b]} \in \mathbb{R}^{m \times T}$ and $b^{[a,b]} \in \mathbb{R}^m$. Assume that $D^{[a,b]}$ is bounded. While this work only considers power output and spinning reserves, the methods used throughout this paper can accommodate any number of services so long as the set of all services remains a bounded polyhedron when the commitment status is fixed.

To demonstrate, consider the most common description of $D^{[a,b]}$ found in the power systems literature, which deals with the following types of constraints: minimum/maximum output, maximum ramping, and start-up/shut-down levels. The constraints defining the polytope $D_{typical}^{[a,b]}$ are:

$$p_i^{[a,b]} \leq 0 \quad \forall i < a \text{ and } i > b \quad (13a)$$

$$\bar{p}_i^{[a,b]} \leq 0 \quad \forall i < a \text{ and } i > b \quad (13b)$$

$$-p_i^{[a,b]} \leq -P \quad \forall i \in [a, b] \quad (13c)$$

$$p_i^{[a,b]} \leq \bar{p}_i^{[a,b]} \quad \forall i \in [a, b] \quad (13d)$$

$$\bar{p}_i^{[a,b]} \leq \min(\bar{P}, SU + (i - a) RU, SD + (b - i) RD) \quad \forall i \in [a, b] \quad (13e)$$

$$\bar{p}_i^{[a,b]} - \bar{p}_{i-1}^{[a,b]} \leq \min(RU, SU + (b - i) RD - P) \quad \forall i \in [a + 1, b] \quad (13f)$$

$$p_i^{[a,b]} - p_{i-1}^{[a,b]} \leq \min(RD, SU + (i - a) RU - P) \quad \forall i \in [a + 1, b]. \quad (13g)$$

Constraints (13a) and (13b) specify that the generator does not output power nor provide reserves while off; (13c) specifies the minimum level of power output when the generator is on. (13d) ensures the power available is at least the power committed. Constraint (13e) enforces the upper bound on power output at time $i$. This ensures the generator does not produce more power than its maximum output $\bar{P}$, the power level it could ramp up to by time $i$ (SU + (i –
a) $RU$, or ramp down from at time $i$ $(SD + (b - i)RD)$. The ramp up constraint (13f) ensures the power jump between times $i - 1$ and $i$ is no more than $RU$ or that which we could ramp back down to in the remaining time $(SU + (b - i)RD - P)$. The ramp down constraints (13g) work symmetrically.

Recalling $UT$ and $DT$ are the minimum up and down time for the generator, let $T$ contain all intervals $[a, b]$ where $1 \leq a \leq a + UT \leq b \leq T$. We also need to consider cases when the generator has been turned on prior to time zero and cases where the generator will be on past time $T$. To account for this, we let the interval $[0, b]$ represent cases where the generator was already on before the planning period and is turned off at time $b$. It is not necessary for $b + 1$ to be larger than $UT$. Similarly, we let the interval $[a, T + 1]$ represent the case where the generator continues to be on after the planning period, where the actual shut-down time is undetermined. Note all polytopes $D_{typical}^{[a,b]}$ are nonempty. By combining the polytopes $D_{typical}^{[a,b]}$ such that the intervals only overlap in feasible combinations, Frangioni and Gentile [13] develop a dynamic-programming approach for scheduling a single generator in polynomial time. We will use the polytopes $D_{typical}^{[a,b]}$ in a similar fashion to develop an extended formulation for the ramping polytope.

### 2.3 Packing Dispatch Polytopes

To develop the extended formulation, we construct an interval graph from $T$, where two intervals $[a, b], [c, d]$ are defined to overlap if $[a, b + DT] \cap [c, d + DT] \neq \emptyset$. That is, $G = (V, E)$ has $V = T$ and edges between two vertices if they overlap, is an interval graph by construction, and hence $G$ is a line graph. We now consider packing the vertices of $G$, that is, selecting a subset of $V_P \subseteq V$ such that for any $u, v \in V_P$, $(u, v) \notin E$. If we use variables $\gamma \in \{0, 1\}^{|T|}$ to indicate whether a vertex (interval) is in the packing or not, then is it well known (since $G$ is a line graph) that the clique inequalities (along with non-negativity) give a convex hull description of the vertex packing problem. That is, the vertices of

$$\Gamma = \left\{ \sum_{\{[a,b] \in T \mid i \in [a, b + DT]\}} \gamma_{[a,b]} \leq 1 \quad i \in [T] \right\}$$

$$\gamma_{[a,b]} \geq 0 \quad \forall [a, b] \in T$$

(14)

are binary and represent all feasible vertex packings. Using the dispatch polytopes developed in Section 2.2, we can write down an extended formulation for a ramping-constrained generator.
Theorem 1. The polytope

\[
D := \begin{cases}
A[a,b]p[a,b] + \bar{A}[a,b]\bar{p}[a,b] \leq \gamma[a,b]b[a,b] & \forall [a, b] \in T \\
\sum_{[a,b] \in T} p[a,b] = p \\
\sum_{[a,b] \in T} \bar{p}[a,b] = \bar{p} \\
p[a,b], \bar{p}[a,b] \in \mathbb{R}_+^T & \forall [a, b] \in T \\
\sum_{\{(a,b) \in T \mid i \in [a,b+DT]\}} \gamma[a,b] \leq 1 & i \in [T] \\
\gamma[a,b] \geq 0 & \forall [a, b] \in T
\end{cases}
\]

is a compact (polynomial-sized in \(T\)) formulation for a ramping-constrained generator, and the vertices of \(D\) have integer \(\gamma\).

Remark 1. Not dispatching the generator in time period \([1, T + 1]\) corresponds to having \(\gamma[a,b] = 0\) for all \([a, b] \in T\).

Linear generation costs \(c \in \mathbb{R}^T\) and fixed start-up (and shut-down) costs \(w \in \mathbb{R}^{|T|}\) can be modeled by optimizing the linear function \(c^\top p + w^\top \gamma\) over \(D\). Still, this formulation is large, and is unlikely to be computationally effective within the problem (1). We can, however, use the above formulation to derive a cut-generating linear program for an individual generator, very similar in spirit to the lift-and-project cuts for disjunctive programs [6]. First, however, we will prove Theorem 1 by extending the classical result of Balas on disjunctive programs [3, 4].

3 Constrained Minkowski Sums of Polyhedra

The success of the disjunctive programming as initially laid out by Balas [3, 4] toward the practical solvability of problems involving indicator constraints is clear; see [7] for a recent overview. We consider an extension of Balas’s classical result (Theorem 2), originally given as a lemma in [12], and show it can be used to model constrained Minkowski sums of polyhedra. This will allow us to prove Theorem 1. Note that this result is stronger than we need, but we provide it here for completeness.

The goal of this section is to arrive at a polyhedral representation of constrained Minkowski sums of polyhedra using indicator variables. First we must dispense with some definitions. Scalar multiples and Minkowski sums for sets
in $\mathbb{R}^n$ are defined in their usual way as

\[ \lambda C := \{ \lambda x \mid x \in C \} \]

(16)

\[ C_1 + C_2 := \{ x_1 + x_2 \mid x_1 \in C_1, x_2 \in C_2 \} \]

(17)

For a set $S \subset \mathbb{R}^n$, $\text{conv}(S)$ is the convex hull of $S$ and $\text{cone}(S)$ is the conic hull of $S$. The orthogonal projection of $S \subset \mathbb{R}^n \times \mathbb{R}^p$ onto $\mathbb{R}^n$ is denoted $\text{proj}_x(S) := \{ x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^p \text{ s.t. } (x, y) \in S \}$. A system of linear inequalities $Ax \leq b$ is said to be a perfect formulation of a set $S \subset \mathbb{R}^n$ if $\text{conv}(S) = \{ x \in \mathbb{R}^n \mid Ax \leq b \}$. For a polyhedron $P \subset \mathbb{R}^n$ we say that a polyhedron $Q \subset \mathbb{R}^n \times \mathbb{R}^p$ is an extended formulation of $P$ if $\text{proj}_x(Q) = P$. Such an extended formulation is said to be compact when only a polynomial number of variables and constraints in the size of the input are needed to describe $Q$. For convenience we (again) use the notation $[m] = \{1, \ldots, m\}$ and subscripts to indicate the components of a vector.

Naturally our tools are those of convex analysis [27, 29, 16], with the Minkowski-Weyl theorem for polyhedra [30] playing a lead role. To motivate the framework developed in this section, consider convex combinations of polyhedra.

Suppose we have a collection $P^1, \ldots, P^m$ of nonempty polyhedra, and notice $\text{conv}(\bigcup_{i \in [m]} P^i) = \bigcup \{ \sum_{i=1}^m \gamma_i P^i \mid \sum_{i=1}^m \gamma_i = 1, \gamma \geq 0 \}$. An interesting question is when is such a set closed and polyhedral. Indeed Theorem 9.8 and subsequent corollaries in Rockafellar [27] give sufficient conditions for closedness. Balas [3, 4] provides sufficient conditions for polyhedrality along with an extended formulation for such a set. We restate Balas’s result.

**Theorem 2.** Consider $m$ polyhedra $P^i = \{ x \in \mathbb{R}^n \mid A^i x \leq b^i \}$ and their polyhedral recession cones $R^i = \{ x \in \mathbb{R}^n \mid A^i x \leq 0 \}$ and let $Q^i$ be a (bounded) polytope such that $P^i = Q^i + R^i$. Define the set $S = \text{conv}(\bigcup_{i \in [m]} P^i)$ and polyhedron $P = \text{conv}(\bigcup_{i \in [m]} Q^i) + \text{conv}(\bigcup_{i \in [m]} R^i)$. Then the polyhedron

\[
Y = \begin{cases} 
A^i x^i \leq \gamma_i b^i, & i \in [m] \\
\sum_{i \in [m]} x^i = x \\
\sum_{i \in [m]} \gamma_i = 1 \\
\gamma_i \geq 0, & i \in [m]
\end{cases}
\]

(18)

provides an extended formulation of $P$. If each $P^i, i \in [m]$, is nonempty then $\text{cl}(S) = P$. Additionally, the vertices of
In the context of Theorem 2 we also have the following result from Jeroslow [17] and Corollary 9.8.1 in [27]:

**Theorem 3.** If $P^1, \ldots, P^m$ are all nonempty and have identical recession cones then $S = P$ and so $Y$ provides a polyhedral extended formulation for $S$.

We would like to generalize the above theorems to allow for different combinations of polyhedra. To be precise, suppose $\Gamma$ is a polyhedron in $\mathbb{R}^m$, and consider the set $\bigcup \{ \sum_{i=1}^{m} \gamma_i P^i \mid \gamma \in \Gamma \}$. A natural question is this: can we derive results similar in spirit to that of the preceding theorems? We answer this question in the affirmative, with a few restrictions on $\Gamma$.

To see what some of these restrictions must be, consider the challenges of using indicator variables as in (18). Suppose we have a polyhedron $P$ with a representation $Ax \leq b$. Clearly $\gamma P = \{ x \mid Ax \leq \gamma b \}$ for all $\gamma > 0$. The first issue is for $\gamma < 0$, $\gamma P = \{ x \mid Ax \geq \gamma b \}$. This shows that allowing the sign to switch on $\gamma$ will not allow the easy modeling of inequalities, and therefore we will, without loss of generality, only consider nonnegative indicator variables. Another issue dealing with the discontinuity of $\gamma P$ when $\gamma$ is near 0 is that by definition $0P = \{0\}$ whereas $\{ x \mid Ax \leq 0b \} = \{ x \mid Ax \leq 0 \}$, which is the polyhedral recession cone of $P$. This demonstrates that in a formulation like (18), while the indicator variables $\gamma$ allow for “control” over the finite part of $P$, the recession directions of $P$ are always included. Similarly, if $P$ is empty, the polyhedral recession cone $\{ x \mid Ax \leq 0 \}$ is not, and will be included in a formulation like (18). For ease of exposition we will restrict ourselves to the case when each polyhedron is nonempty, but note that with some extra notation we could extend the results of Section 3.1 to include possibly empty polyhedra.

### 3.1 The Extended Formulation

Now consider the set $S := \bigcup_{\gamma \in \Gamma} \{ \sum_{i=1}^{m} \gamma_i P^i \}$, where $P^i, i \in [m]$, are nonempty polyhedra in $\mathbb{R}^n$ and $\Gamma \subseteq \mathbb{R}^m_+$ is a nonempty, nonnegative polyhedron. The goal is to arrive at a polyhedral representation for $S$. The exposition here follows that found in Section 4.9 of [10].

**Theorem 4.** Consider $m$ nonempty polyhedra $P^i = \{ x \in \mathbb{R}^n \mid A^i x \leq b^i \}, i \in [m]$, and for each $i \in [m]$ let $Q^i$ be a (bounded) polytope in $\mathbb{R}^n$ and $R^i$ be a (closed convex) cone in $\mathbb{R}^n$ such that $P^i = Q^i + R^i$. Let $\Gamma \subseteq \mathbb{R}^m_+$.
be a nonempty polyhedron. Consider the set 

\[ P := \bigcup_{\gamma \in \Gamma} \left( \sum_{i=1}^{m} \gamma_i Q^i + \sum_{i=1}^{m} R^i \right) \]

and consider the polyhedron 

\[ Y \subseteq \mathbb{R}^{n+nm+m} \]

defined by

\[
Y := \left\{ \begin{array}{l}
A^i x^i \leq \gamma_i b^i, \ i \in [m] \\
\sum_{i=1}^{m} x^i = x \\
(\gamma_1, \ldots, \gamma_m) = \gamma \in \Gamma.
\end{array} \right. \tag{19}
\]

Then \( P = \text{proj}_x(Y) := \{ x \in \mathbb{R}^n \mid \exists (x^1, \ldots, x^m, \gamma) \in \mathbb{R}^{nm+m} \text{ s.t. } (x, x^1, \ldots, x^m, \gamma) \in Y \} \). In particular, \( P \) is a polyhedron.

**Proof.** Let \( x \in P \) (\( P \) is nonempty as the union of the sum of nonempty sets). There exists points \( q^i \in Q^i, r^i \in R^i \) and \( \gamma \in \Gamma \) such that \( x = \sum_{i=1}^{m} \gamma_i q^i + \sum_{i=1}^{m} r^i \). Define \( x^i = \gamma_i q^i + r^i \) for \( i \in [m] \). Now by construction \( x = \sum_{i=1}^{m} x^i \) and \( A^i x^i = A^i (\gamma_i q^i + r^i) = \gamma_i A^i q^i + A^i r^i \leq \gamma_i b^i \) for all \( i \in [m] \). Hence \( (x, x^1, \ldots, x^m, \gamma) \in Y \), so \( P \subseteq \text{proj}_x(Y) \).

Conversely, let \((x, x^1, \ldots, x^m, \gamma) \in Y \). Consider \( I^+ := \{ i \mid \gamma_i > 0 \} \) and \( I^0 := \{ i \mid \gamma_i = 0 \} \). For \( i \in I^+ \), \( A^i x^i \leq \gamma_i b^i \) and so \( x^i \in \gamma_i Q^i + R^i \). For \( i \in I^0 \), \( A^i x^i \leq 0 \) and so \( x^i \in R^i = \gamma_i Q^i + R^i \). Since \( x = \sum_{i=1}^{m} x^i \in \sum_{i=1}^{m} (\gamma_i Q^i + R^i) \) and \( \gamma \in \Gamma \), this shows \( x \in P \), and hence \( \text{proj}_x(Y) \subseteq P \).

As the projection of a polyhedron, \( P \) is itself a polyhedron. \( \Box \)

**Remark 2.** For all \( \Gamma \subseteq \mathbb{R}^m_+ \), \( Y \) provides a polynomial-size (in \( \dim(P^i) \) and \( \dim(\Gamma) \)) polyhedral representation of \( P \).

Further, if for all \( i \in [m] \), \( P^i \) is bounded (i.e., \( R^i = \{ 0 \} \)), then \( P = S \) and \( Y \) provides a compact formulation for \( S \).

**Remark 3.** If \( \Gamma \subseteq \mathbb{R}^m_{++} \) (the open, strictly positive orthant), then \( \gamma_i P^i = \gamma_i Q^i + R^i \forall (\gamma_1, \ldots, \gamma_m) \in \Gamma \). Therefore \( P = S \) and so \( Y \) provides a compact formulation for \( S \).

We note that clearly \( S \subseteq P \). The next theorem demonstrates that \( \text{cl}(S) = P \) with a restriction on \( \Gamma \).

**Theorem 5.** Let \( \Gamma \subseteq \mathbb{R}^m_+ \) and \( P^1, \ldots, P^m \subseteq \mathbb{R}^n \) be nonempty polyhedra. Suppose there exists \( \gamma \in \Gamma \) such that \( \gamma_i > 0 \forall i \in [m] \). Then for \( P \) and \( S \) defined as above, \( \text{cl}(S) = P \).

**Proof.** First consider \( \text{cl}(S) \subseteq P \). Since \( P \) as a polyhedron is closed, it suffices to show \( S \subseteq P \). Hence let \( x \in S \). Then \( \exists \gamma \in \Gamma, p^i \in P^i \) for \( i \in [m] \) such that \( x = \sum_{i=1}^{m} \gamma_i p^i \). As above for each \( i \in [m] \), consider \( P^i = Q^i + R^i \),
so for each $i \in [m]$ we have $p^i = q^i + r^i$ for $q^i \in Q^i$ and $r^i \in R^i$. Thus $x = \sum_{i=1}^{m} \gamma_i q^i + \sum_{i=1}^{m} \gamma_i r^i$, and since $\gamma_i q^i \in \gamma_i Q^i$ and $\gamma_i r^i \in R^i$ (as $R^i$ is a closed convex cone, $\gamma_i \geq 0$), we have $x \in \mathcal{P}$.

Conversely, let $x \in \mathcal{P}$. Then there exists $\gamma \in \Gamma$, $q^i \in Q^i$, and $r^i \in R^i$ such that $x = \sum_{i=1}^{m} \gamma_i q^i + \sum_{i=1}^{m} \gamma_i r^i$. By assumption $\exists \hat{\gamma} \in \Gamma$ that is strictly positive. By convexity, $(1 - \varepsilon)\gamma + \varepsilon \hat{\gamma} \in \Gamma$ for $\varepsilon \in (0, 1)$; further $(1 - \varepsilon)\gamma + \varepsilon \hat{\gamma} > 0$ for $\varepsilon \in (0, 1)$. Define $x^\varepsilon := \sum_{i=1}^{m} [(1 - \varepsilon)\gamma_i + \varepsilon \hat{\gamma}_i]q^i + \sum_{i=1}^{m} \gamma_i r^i$. Clearly $\lim_{\varepsilon \to 0^+} x^\varepsilon = x$, and we see that $x^\varepsilon = \sum_{i=1}^{m} [(1 - \varepsilon)\gamma_i + \varepsilon \hat{\gamma}_i](q^i + r^i)/(1 - \varepsilon)\gamma_i + \varepsilon \hat{\gamma}_i)$. Since $q^i + r^i/(1 - \varepsilon)\gamma_i + \varepsilon \hat{\gamma}_i) \in \mathcal{P}^i$ for $i \in [m]$, $\varepsilon \in (0, 1)$ and $(1 - \varepsilon)\gamma_i + \varepsilon \hat{\gamma} \in \Gamma$ for $\varepsilon \in (0, 1)$, $x^\varepsilon \in \mathcal{S} \forall \varepsilon \in (0, 1)$. Hence $x \in \overline{\text{cl}}(\mathcal{S})$.

The requirement that $\Gamma$ have a strictly positive element should not be seen as overly restrictive. If for some $i$, $\gamma_i = 0 \forall \gamma \in \Gamma$, then we should probably discard this particular $\mathcal{P}^i$ since it never contributes to the sum.

**Remark 4.** If there exists $\hat{\gamma} \in \Gamma$ such that $\hat{\gamma} > 0$ and $\mathcal{P}^1, \ldots, \mathcal{P}^m$ are all nonempty, then Theorems 4 and 5 together imply that $\overline{\text{cl}}(\mathcal{S}) = \text{proj}_x(Y)$.

**Theorem 6.** Suppose $\mathcal{P}^1, \ldots, \mathcal{P}^m$ are nonempty polyhedra with identical recession cones, and $\Gamma \subset \mathbb{R}^m_+$ is a polyhedron such that $0 \notin \Gamma$. Then $\mathcal{S} = \bigcup_{\gamma \in \Gamma} \left( \sum_{i=1}^{m} \gamma_i \mathcal{P}^i \right)$ is a polyhedron and $\mathcal{S} = \text{proj}_x(Y)$.

**Proof.** Let $x \in \mathcal{P}$. Then there exists $q^i \in Q^i$, $r^i \in R^i$ and $\gamma \in \Gamma$ such that $x = \sum_{i=1}^{m} \gamma_i q^i + \sum_{i=1}^{m} \gamma_i r^i$. By assumption there exist $j \in [m]$ such that $\gamma_j > 0$. As the $\mathcal{P}^i$’s have identical recession cones, we have $\sum_{i=1}^{m} \gamma_i r^i \in \gamma_j \mathcal{P}^j$. Define $p^j = q^i + \sum_{i=1}^{m} \gamma_j r^i$ and $p^i = q^i$ for $i \neq j$, and it follows that $x = \sum_{i=1}^{m} \gamma_i p^i$. Hence $x \in \mathcal{S}$. The result then follows from Theorem 4. \qed

**Remark 5.** To see the necessity of $0 \notin \Gamma$, consider the sets $\mathcal{S}$ and $\mathcal{P}$ when $\gamma = 0$. If $\mathcal{P}^1, \ldots, \mathcal{P}^m$ have the same recession cone $R$, we see that $\mathcal{P}|_{\gamma=0} = \sum_{i=1}^{m} 0Q^i + \sum_{i=1}^{m} R^i = R$, whereas $\mathcal{S}|_{\gamma=0} = \sum_{i=1}^{m} 0\mathcal{P}^i = \{0\}$, and $R = \{0\}$ if and only if all the $\mathcal{P}^i$’s are bounded. Hence we can do away with the assumption $0 \notin \Gamma$ in Theorem 6 if all the $\mathcal{P}^i$’s are bounded.

It may be that $\Gamma$ is the continuous relaxation of some integer set which determines the polyhedra $\mathcal{P}^i$ simultaneously allowed in the sum. The next theorem shows that vertices and extreme rays of $Y$ have $\gamma$ components which are vertices and extreme rays of $\Gamma$, hence if $\Gamma$ is a perfect formulation for some integer set, vertices of $Y$ will have integer $\gamma$. 

---

[13]
Further, even if $\Gamma$ is not a perfect formulation, this shows that to find solutions with integer $\gamma$ one need only consider cuts on $\Gamma$ and not the entire polyhedron $Y$. For ease of notation, for $y \in Y$ define $y_r$ to be the components of $y$ in $\Gamma$.

Finally, we note that a version of Theorem 7 appears as Lemma 5 in [12], although it is restricted to the pure integer case, and the proof is merely sketched. We provide a complete proof and drop any assumption of integrality.

**Theorem 7.** $Y = \text{conv}(V) + \text{cone}(R)$, for finite sets $V$ and $R$, where for each vertex $v \in V$, $v_r$ is a vertex of $\Gamma$ and for each extreme ray $r \in R$, $r_r$ is an extreme ray of $\Gamma$. That is, $\text{proj}_r(Y) = \Gamma$.

**Proof.** Let $y \in Y$ such that $y = (x, x_1, \ldots, x_m, \gamma_1, \ldots, \gamma_m)$ and define $\gamma := y_r$. Since $\Gamma$ is a polyhedron, by the Minkowski-Weyl theorem there exist vectors $v^1, \ldots, v^p, r^1, \ldots, r^q \in \mathbb{R}^m$ and $\lambda \in \mathbb{R}^p_+, \mu \in \mathbb{R}^q_+$ such that $\gamma = \sum_{k=1}^p \lambda_k v^k + \sum_{l=1}^q \mu_l r^l$ and $\sum_{k=1}^p \lambda_k = 1$. In particular, we have

$$\gamma_i = \sum_{k=1}^p \lambda_k v^k_i + \sum_{l=1}^q \mu_l r^l_i, \text{ with } \sum_{k=1}^p \lambda_k = 1, \forall i \in [m] \quad (20)$$

Let $I_+ = \{i \mid \gamma_i > 0\}$ and $I_0 = \{i \mid \gamma_i = 0\}$. Define

$$x^k_i := \begin{cases} 
   x_i v^k_i / \gamma_i & \text{if } i \in I_+ \\
   x_i & \text{if } i \in I_0 \text{ and } \lambda_k > 0 \\
   \hat{x}_i^k & \text{if } i \in I_0 \text{ and } \lambda_k = 0
   \end{cases} \quad \forall k \in [p], \ x^l_i := \begin{cases} 
   x_i r^l_i / \gamma_i & \text{if } i \in I_+ \\
   0 & \text{if } i \in I_0 \text{ and } \mu_l > 0 \\
   \hat{x}_i^l & \text{if } i \in I_0 \text{ and } \mu_l = 0
   \end{cases} \quad \forall l \in [q], \quad (21)$$

and $x^k = \sum_{i=1}^m x^k_i$ for $k \in [p]$ and $x^l = \sum_{i=1}^m x^l_i$ for $l \in [q]$. For $k \in [p]$ define $y^k := (x^k, x^k_1, \ldots, x^k_m, v^k_1, \ldots, v^k_m)$ and for $l \in [q]$ define $y^l := (x^l, x^l_1, \ldots, x^l_m, r^l_1, \ldots, r^l_m)$.

We first check the feasibility of the points constructed above. So for each $k \in [p]$, consider $y^k$. By construction $y^k_1 \in \Gamma$ and $x^k = \sum_{i=1}^m x^k_i$, so for feasibility we need verify that $A^i x^k_i \leq v^k_i b^i$. Suppose $i \in I_+$, then $A^i x_i \leq \gamma_i b^i$, and multiplying both sides by $v^k_i$ and dividing by $\gamma_i$ shows $x^k_i$ is feasible. Now suppose $i \in I_0$ and so $A^i x_i \leq 0$. If $\lambda_k > 0$, then we must have $v^k_i = 0$, so $x^k_i$ is feasible. If $\lambda_k = 0$, $x^k_i$ is feasible by construction (since each $P^i$ is nonempty we can always find such a point $\hat{x}_i^k$). The feasibility of $y^l$ for each $l \in [q]$ is similar.

Now we need show $y = \sum_{k=1}^p \lambda_k y^k + \sum_{l=1}^q \mu_l y^l$ to complete the proof. So first suppose $i \in I_+$, then

$$\sum_{k=1}^p \lambda_k x^k_i + \sum_{l=1}^q \mu_l x^l_i = \sum_{k=1}^p \lambda_k x^k_i / \gamma_i + \sum_{l=1}^q \mu_l x^l_i / \gamma_i = \frac{x_i}{\gamma_i} (\sum_{k=1}^p \lambda_k v^k_i + \sum_{l=1}^q \mu_l r^l_i) = x_i. \quad \text{Conversely, suppose } i \in I_0, \text{ then } \sum_{k=1}^p \lambda_k x^k_i + \sum_{l=1}^q \mu_l x^l_i = \sum_{k: \lambda_k > 0} \lambda_k x_i + \sum_{k: \lambda_k = 0} \lambda_k \hat{x}_i^k + \sum_{l: \mu_l > 0} \mu_l 0 +$$
\[
\sum l \mu l = 0 \quad \mu l = 0 \quad l i = \sum k : \lambda k > 0 \lambda k = x_i. \]

It then follows, \( \sum_{k=1}^{p} \lambda k x_k^k + \sum_{l=1}^{q} \mu l x_l^l = \sum_{k=1}^{p} \lambda k \sum_{i=1}^{m} x_k^k + \sum_{l=1}^{q} \mu l \sum_{i=1}^{m} x_l^l = \sum_{i=1}^{m} (\sum_{k=1}^{p} \lambda k x_k^k + \sum_{l=1}^{q} \mu l x_l^l) = \sum_{i=1}^{m} x_i = x. \) Hence, we have shown \( y = \sum_{k=1}^{p} \lambda k y_k^k + \sum_{l=1}^{q} \mu l y_l^l \) with \( \lambda, \mu \geq 0 \) and \( \sum_{k=1}^{p} \lambda k = 1 \), proving the theorem.

As mentioned, Theorem 7 demonstrates that if \( \Gamma \) is a perfect formulation of some integer set and the variables \( x^i \) are continuous, then \( Y \) (under the given assumptions) provides a perfect formulation for \( S \big|_{Z^+} = \bigcup_{\gamma \in \Gamma \cap Z^+} \{ \sum_{i=1}^{m} \gamma i P_i \} \).

Noting that the polyhedron of Theorem 1 is exactly of this form, we see that the vertices of the polytope \( D \) must have integer \( \gamma \). Further, \( D \) represents the feasible dispatch \( \Pi^{g} \) for a ramping constrained generator. Although \( D \) is polynomially large, we will use this to develop a procedure for generating cuts based on the polytope \( D \) similar in spirit to lift-and-project cuts \([6]\). We do so by lifting solutions in \( \Pi^{g} 3\text{-bin} \) to the “\( D \)-space”, getting a separating cut, and projecting the generated cut back into 3-bin space.

### 4 A Cutting-Plane Procedure for the 3-bin Formulation

Recalling now the typical 3-bin formulation described in Section 2.1 and the new extended formulation developed in Sections 2.2 and 2.3 we see how these formulations can be “connected” through a linear transformation, which will be the basis for our cut-generation routine.

#### 4.1 From Dispatch Polytope Space to 3-bin Space

Dropping the superscript \( g \) for a moment to focus on one generator, recall equation (15). Although it is clear from the formulation of \( D \) how to project it on to the space of \((p, \bar{p})\) variables, by using a linear transformation we can project \( D \) into 3-bin space, that is, the space of the \((p, \bar{p}, u, v, w)\) variables from Section 2.1. Of course, \( p \) and \( \bar{p} \) remain the
same, and we can link \(\gamma\) and \((u, v, w)\) as follows:

\[
\sum_{\{[a, b] \in \mathcal{T} \mid i \in [a, b]\}} \gamma_{[a, b]} = u_i \quad i \in [T] \tag{22a}
\]

\[
\sum_{\{[a, b] \in \mathcal{T} \mid i = a\}} \gamma_{[a, b]} = v_i \quad i \in [T] \tag{22b}
\]

\[
\sum_{\{[a, b] \in \mathcal{T} \mid i = b+1\}} \gamma_{[a, b]} = w_i \quad i \in [T]. \tag{22c}
\]

Notice by adding the constraints (22) to the formulation of \(D\) (15), for a given 3-bin solution \((p^*, \overline{p}^*, u^*, v^*, w^*)\) \(\in \mathbb{R}^{\Pi_{3\text{-bin}}}\), either the system will be feasible, in which case this 3-bin solution is in the ramping polytope, or the system will not be feasible, in which case this 3-bin solution is not in the ramping polytope. In the latter case we can use the infeasibility information to generate a cut for the 3-bin space which will cutoff this infeasible schedule. This is the basis for our cut generating linear program. Now define the projection of \(D\) into 3-bin space, \(D^{\Pi_{3\text{-bin}}}\), in the usual manner:

\[
D^{\Pi_{3\text{-bin}}} := \{(p, \overline{p}, u, v, w) \in \mathbb{R}_+^{5T} \mid \forall [a, b] \in \mathcal{T}, \exists p^{[a, b]}, \overline{p}^{[a, b]} \in \mathbb{R}_+^{T}, \gamma_{[a, b]} \in \mathbb{R}_+ \text{ satisfying (15) and (22)}\}. \tag{23}
\]

Clearly \(\text{conv}(\Pi_{3\text{-bin}}) \subseteq D^{\Pi_{3\text{-bin}}} \subseteq \mathbb{R}^{\Pi_{3\text{-bin}}\text{-bin}},\) and since \(D^{\Pi_{3\text{-bin}}}\) is the projection of an integer polytope, it follows that \(\text{conv}(\Pi_{3\text{-bin}}) = D^{\Pi_{3\text{-bin}}}[5]\).

### 4.2 A Cut-Generating Linear Program

For ease we will consider the dual form of the cut-generating LP, which is derived from (15) and (22) above. Let \(e\) be the appropriately sized vector of 1’s and suppose \(z \in \mathbb{R}_+.\) Let \((p^*, \overline{p}^*, u^*, v^*, w^*)\) be a solution vector in 3-bin space.

Consider the following linear program:
\[
\min z \\
\text{subject to} \\
\pi A^{[a,b]} p^{[a,b]} + A^{[a,b]} \bar{p}^{[a,b]} \leq \gamma_{[a,b]} b^{[a,b]} + ze \\
\forall [a, b] \in \mathcal{T} \tag{24a}
\]

\[
\delta \sum_{\{[a, b] \in \mathcal{T} \mid i \in [a, b + DT]\}} \gamma_{[a, b]} \leq 1 + z \\
i \in [T] \tag{24b}
\]

\[
\varepsilon \sum_{[a, b] \in \mathcal{T}} p^{[a,b]} = p^* \\
\tag{24c}
\]

\[
\mu \sum_{[a, b] \in \mathcal{T}} \bar{p}^{[a,b]} = \bar{p}^* \\
\tag{24d}
\]

\[
\xi \sum_{\{[a, b] \in \mathcal{T} \mid i \in [a, b]\}} \gamma_{[a, b]} = u_i^* \\
i \in [T] \tag{24e}
\]

\[
\alpha \sum_{\{[a, b] \in \mathcal{T} \mid i = a\}} \gamma_{[a, b]} = u_i^* \\
i \in [T] \tag{24f}
\]

\[
\sigma \sum_{\{[a, b] \in \mathcal{T} \mid i = b + 1\}} \gamma_{[a, b]} = w_i^* \\
i \in [T] \tag{24g}
\]

\[
z \in \mathbb{R}_+; \\ p^{[a,b]}, \bar{p}^{[a,b]} \in \mathbb{R}^T_+, \gamma_{[a,b]} \in \mathbb{R}_+, \\
\forall [a, b] \in \mathcal{T}, \tag{24h}
\]

where \( \pi \in \mathbb{R}_0^{m|\mathcal{T}|} \) is the set of dual variables for constraints (24b), \( \delta \in \mathbb{R}^T \) is the set of dual variables for (24c), and \( \varepsilon, \mu, \xi, \alpha, \sigma \in \mathbb{R}^T \) are the sets of dual variables for constraints (24d - 24h), respectively. We observe if \( z^* \), the optimal value of (24), is 0, then \( (p^*, \bar{p}^*) \) is a feasible production schedule for \( D \), and if not, we can use the optimal dual vector to cut off the 3-bin solution \( (p^*, \bar{p}^*, u^*, v^*, w^*) \in R^3_{\Pi_{3-bin}} \). To demonstrate, suppose \( z^* > 0 \) and we have an optimal dual vector \( \pi^*, \delta^*, \varepsilon^*, \mu^*, \xi^*, \alpha^*, \sigma^* \). Then by strong duality \( z^* = (\delta^*)^T e + (\varepsilon^*)^T p^* + (\mu^*)^T \bar{p}^* + (\xi^*)^T u^* + (\alpha^*)^T v^* + (\sigma^*)^T w^* \geq 0 \), and so the cut \( (\delta^*)^T e + (\varepsilon^*)^T p + (\mu^*)^T \bar{p} + (\xi^*)^T u + (\alpha^*)^T v + (\sigma^*)^T w \leq 0 \) cuts off the solution \( (p^*, \bar{p}^*, u^*, v^*, w^*) \) in 3-bin space (that is, it is a valid separating hyperplane between \( (p^*, \bar{p}^*, u^*, v^*, w^*) \) and \( D^3_{\Pi_{3-bin}} \)). Further, by choosing the optimal such cut (instead of any dual feasible solution to (24)), we are maximizing the depth of the cut.
4.3 Implementation

To test the efficacy of these cuts, we implement them as a callback for a utility-scale unit commitment problem based on the set of FERC generators [18]. We use the standard 3-bin formulation for the master unit commitment MIP, as discussed in Section 2.1, that is:

\[
\begin{align*}
\min & \quad \sum_{g \in G} \sum_{t \in [T]} \left( \sum_{l \in [L]} (l^g l^g_t) + C^g C^g_t + H^g H^g_t + R^g u^g_t + D^g w^g_t \right) \\
\text{s.t.} & \quad \sum_{g \in G} p^g_t \geq D_t, \forall t \in [T] \\
& \quad \sum_{g \in G} \bar{p}^g_t \geq D_t + R_t, \forall t \in [T] \\
& \quad (10), (11) \ \forall g \in \mathcal{G}, \\
& \quad (p^g, \bar{p}^g, u^g, v^g, w^g) \in \Pi^g_{3\text{-}bin}, \ \forall g \in \mathcal{G},
\end{align*}
\]  

(25a)

(25b)

(25c)

(25d)

(25e)

We only consider cuts on a subset of the generators, namely for those that have binding ramping constraints while operating and those that have a minimum run time of at least 2. That is, we consider cuts on \( G^C := \{ g \in \mathcal{G} \mid (P^g - \bar{P}^g) > \min\{RD^g, RU^g\} \text{ and } UT^g \geq 2 \} \). Thus we implement the cuts developed in a callback, namely, given the current LP relaxation for (25), for each generator in \( G^C \), we use (24) to determine if \((p^*, \bar{p}^*, u^*, v^*, w^*) \in D^g \Pi^g_{3\text{-}bin}\), and if not we add the violated ramping inequality given by the dual vector of (24). We generate the cuts using the “bundling” approach from [6], that is, at the current master LP relaxation we try to generate a cut for each \( g \in G^C \) and give these to the solver together.

Next, we discuss some computational enhancements to this general outline. First, to mitigate numerical issues, we only add cuts for which \( z^* \geq 10^{-2} \). Because of this we use the artificial lower bound \( z \geq 10^{-2} \) so that the simplex algorithm will terminate if we have determined any cut generated may be numerically unstable. Second, after a round of calls to the cut-generating LPs we use the existing basis information if and only if we did not generate a cut from it in the previous pass. The intuition is that if we did not generate a cut from this generator previously then the current 3-bin vector is probably close to the previous one. On the other hand, if we did generate a cut, then (we hope) the current 3-bin vector is far away from the previous one, so we discard the previous basis information. Additionally, we can make an enhancement based on symmetry by observing that if \( g_1, g_2 \in G^C \) have identical parameters, a cut
generated for \( g_1 \) is valid for \( g_2 \), and vice versa. That is, if we denote generators identical to \( g \) as \( \text{orb}(g) \), for every cut generated for \( g \) we add the associated cut for every \( \hat{g} \in \text{orb}(g) \). Finally, we choose an aggressive branch-and-cut strategy, generating cuts at the root node, for the first 50 nodes, and then every 100 nodes thereafter.

### 4.4 Computational Tests

All computational tests were performed on a Dell PowerEdge T620 with 2 Intel Xeon E5-2670 processors and 256GB of memory running Ubuntu 14.04.2. Gurobi 6.5.0 was used as the MIP and LP solver for all problems, and the callback routine was implemented using Gurobi’s Python interface. For all problems the number of available threads was set to 1. For the MIP unit commitment problem, the parameter PreCrush was set to 1 to facilitate adding cuts in callbacks along with a time limit of 1800 seconds. All other parameters were preserved at default. A dummy callback was used for instances where cuts were not added.

To generate a diverse set of unit commitment test instances, real time load, day-ahead reserves, and wind generation for 2015 were obtained from PJM’s website [1, 2]. For each day in 2015 a 24-hour unit commitment problem was formulated, with wind generation accounted for as negative demand in (25). For ease, the daylight savings days of 08 Mar and 01 Nov were excluded. 31 Dec was excluded for lack of available data. For the months of April - September the set of summer generators was used, and the winter generators were considered for the remaining six months. Generators with missing cost curves were excluded, and generators with missing up/down time data were given \( UT^g = DT^g = 1 \). Generators marked as wind powered were dropped as wind generation is considered separately. In total then 935 generators were considered for the winter system and 978 generators were considered for the summer system. Given the selection criteria above for \( \mathcal{G}^C \), \( |\mathcal{G}^C| = 459 \) for the winter system and \( |\mathcal{G}^C| = 492 \) for the summer system. As no data on start-up or shut-down ramps are provided, \( SU^g = SD^g = P^g \) for all \( g \in \mathcal{G} \). Additionally, no data on cool down is provided, so we assume all generators cool down in twice their minimum down time period, i.e., \( DT^g_C = 2DT^g \). We use the data provided on initial status and assume all generators currently on are available to be turned off and operating at minimum power.
<table>
<thead>
<tr>
<th></th>
<th>No User Cuts</th>
<th>Ramping Polytope Cuts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (s)</td>
<td>Nodes</td>
</tr>
<tr>
<td>Mean</td>
<td>186.01</td>
<td>9.98</td>
</tr>
<tr>
<td>Min</td>
<td>85.48</td>
<td>0</td>
</tr>
<tr>
<td>Median</td>
<td>168.97</td>
<td>0</td>
</tr>
<tr>
<td>Max</td>
<td>557.82</td>
<td>715</td>
</tr>
<tr>
<td>Wins</td>
<td>140</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 1: Initial Computational Results

4.5 Initial Results

Of the 362 instances tested, 361 were feasible, and the summary statistics for these instances with and without the cuts developed above are given in Table 1. As we can see, none of the problems in the test set devised are particularly difficult for a modern MIP solver, though there may be some slight benefit to adding the cuts. Most instances, however, are solved at the root node, and no instance takes more than about 10 minutes or 1000 nodes. It is worth noting for these instances that Gurobi needs quite a bit of time (usually 60-120 seconds) to solve the root relaxation, and then spends quite a bit of time at the root node generating cuts and applying heuristics. In the “Wins” row we report the number of instances for which that method did strictly better than the opposing method. Here we see the cuts usually result in a better time, though just slightly, and in most cases ($n = 310$) both methods need the same number of nodes to prove optimality to the default tolerance of 0.01%.

4.6 High Wind Instances

To create more difficult test instances, we again used the 2015 data from PJM, but considered increased wind penetration. In 2015, wind energy accounted for approximately 2% of energy demanded. A recent study conducted for PJM suggested that the interconnection could handle renewable penetration as high as 30%, which may be coming online as soon as 2026 [15]. Therefore, to create high-wind penetration instances, we multiplied the 2015 wind data by a factor of 15 to get to 30% wind energy. Note that our model implicitly allows for the possibility of curtailment (as we...
consider wind as negative load); further, if the wind is greater than load at a given hour it may also provide reserves. Given the greater swings in the net-load curve that the extra wind generation causes, we would expect these instances to be much harder than the base-case instances, and indeed, we find this to be the case.

Of the 362 instances tested, 6 timed out for both methods, and 356 solved for both methods (all instances were feasible). The summary statistics for the instances which did not time out are reported in Table 2. As we can see, there is a 28% mean reduction in nodes, and a modest reduction in mean solve time. To see the impact on more interesting instances, those for which either method took more than 10 minutes to solve (but did not time out) are detailed in Table 3. For these harder instances we can see that for the most part the cuts are effective at reducing the enumeration necessary to arrive at and prove an optimal solution. We have a mean reduction in run time of about 200 seconds, such that the typical hard instance went from taking approximately 14 minutes to 11 minutes to solve. As these problems are usually solved in a 10 or 15 minute time window, this is a significant improvement. Additionally, there is a 39% reduction in mean nodes for these instances, suggesting that strengthening the feasible region for the ramping-constrained generators with cuts from the ramping polytope eliminates some enumeration. Lastly in Table 4 we summarize the 6 instances which timed out, reporting the final MIP gap in place of computational time. There do not seem to be any conclusions that can be safely drawn from these 6 instances.

<table>
<thead>
<tr>
<th></th>
<th>No User Cuts</th>
<th></th>
<th>Ramping Polytope Cuts</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (s)</td>
<td>Nodes</td>
<td>Time (s)</td>
<td>Nodes</td>
</tr>
<tr>
<td>Mean</td>
<td>234.02</td>
<td>153.35</td>
<td>221.42</td>
<td>109.65</td>
</tr>
<tr>
<td>Min</td>
<td>58.40</td>
<td>0</td>
<td>55.88</td>
<td>0</td>
</tr>
<tr>
<td>Median</td>
<td>197.41</td>
<td>0</td>
<td>191.81</td>
<td>0</td>
</tr>
<tr>
<td>Max</td>
<td>1523.65</td>
<td>9875</td>
<td>999.64</td>
<td>6260</td>
</tr>
<tr>
<td>Wins</td>
<td>159</td>
<td>55</td>
<td>197</td>
<td>79</td>
</tr>
<tr>
<td>Cuts Added</td>
<td></td>
<td></td>
<td></td>
<td>676.55</td>
</tr>
<tr>
<td>Cut Time (s)</td>
<td></td>
<td></td>
<td></td>
<td>15.81</td>
</tr>
</tbody>
</table>

Table 2: High Wind Computational Summary, Solved Instances
<table>
<thead>
<tr>
<th>Date</th>
<th>No User Cuts</th>
<th>Ramping Polytope Cuts</th>
<th>Ramping Polytope Cuts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (s)</td>
<td>Nodes</td>
<td>Time (s)</td>
</tr>
<tr>
<td>04 Jan</td>
<td>601.93</td>
<td>2841</td>
<td>549.02</td>
</tr>
<tr>
<td>15 Mar</td>
<td>988.85</td>
<td>4129</td>
<td>767.58</td>
</tr>
<tr>
<td>01 Apr</td>
<td>704.16</td>
<td>554</td>
<td>451.77</td>
</tr>
<tr>
<td>03 Apr</td>
<td>644.06</td>
<td>689</td>
<td>556.55</td>
</tr>
<tr>
<td>12 Apr</td>
<td>742.06</td>
<td>1310</td>
<td>572.83</td>
</tr>
<tr>
<td>15 Apr</td>
<td>1523.65</td>
<td>627</td>
<td>480.67</td>
</tr>
<tr>
<td>25 Apr</td>
<td>1152.09</td>
<td>1440</td>
<td>893.04</td>
</tr>
<tr>
<td>12 May</td>
<td>467.61</td>
<td>354</td>
<td>632.28</td>
</tr>
<tr>
<td>24 May</td>
<td>988.10</td>
<td>686</td>
<td>378.65</td>
</tr>
<tr>
<td>23 Jun</td>
<td>723.91</td>
<td>583</td>
<td>695.87</td>
</tr>
<tr>
<td>02 Oct</td>
<td>620.03</td>
<td>5030</td>
<td>463.35</td>
</tr>
<tr>
<td>21 Nov</td>
<td>1193.51</td>
<td>4125</td>
<td>946.77</td>
</tr>
<tr>
<td>25 Nov</td>
<td>567.15</td>
<td>2648</td>
<td>600.60</td>
</tr>
<tr>
<td>16 Dec</td>
<td>993.68</td>
<td>5946</td>
<td>999.64</td>
</tr>
<tr>
<td>23 Dec</td>
<td>793.52</td>
<td>9875</td>
<td>662.65</td>
</tr>
<tr>
<td>Mean</td>
<td>846.95</td>
<td>2722.5</td>
<td>643.42</td>
</tr>
</tbody>
</table>

Table 3: High Wind, Harder Instances
<table>
<thead>
<tr>
<th>Date</th>
<th>MIP Gap</th>
<th>Nodes</th>
<th>MIP Gap</th>
<th>Nodes</th>
<th>Cuts Added</th>
<th>Cut Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 Oct</td>
<td>0.0102%</td>
<td>10202</td>
<td>0.0107%</td>
<td>10202</td>
<td>2768</td>
<td>28.55</td>
</tr>
<tr>
<td>12 Nov</td>
<td>0.1287%</td>
<td>2613</td>
<td>0.0870%</td>
<td>3350</td>
<td>2313</td>
<td>32.44</td>
</tr>
<tr>
<td>14 Nov</td>
<td>0.0111%</td>
<td>10202</td>
<td>0.0120%</td>
<td>5263</td>
<td>4048</td>
<td>44.35</td>
</tr>
<tr>
<td>17 Nov</td>
<td>0.1015%</td>
<td>2242</td>
<td>0.0938%</td>
<td>1771</td>
<td>3767</td>
<td>31.95</td>
</tr>
<tr>
<td>26 Nov</td>
<td>0.2110%</td>
<td>4778</td>
<td>0.1930%</td>
<td>4565</td>
<td>1717</td>
<td>16.66</td>
</tr>
<tr>
<td>20 Dec</td>
<td>0.0232%</td>
<td>10202</td>
<td>0.0210%</td>
<td>10202</td>
<td>2775</td>
<td>35.07</td>
</tr>
</tbody>
</table>

Table 4: High Wind, Timed Out Instances

### 4.7 Reflections

It is worth taking a moment to bridge the gap between the computation results presented in this section and those reported on 2 and 3-period ramping inequalities recently, namely [11] and [25]. We note that the cuts given by (24) are a superset of those presented in these two papers. The “slow-start” generators in [11] take an average of 4 time periods to ramp from $SU$ to $\mathcal{P}$, and the “fast-start” generators need an average of 3 time periods. Contrarily, for both the winter and summer generator sets, the generators in $\mathcal{G}^C$ take an average of just 2 time periods to ramp from $SU$ to $\mathcal{P}$. Similarly, in the instances from [9, 23] used in [25], every generator in the test set needs 4 time periods to ramp up to full power once on. This test set also contains large amounts of symmetry, which for unit commitment is not perfectly encoded in the formulation symmetry, and hence cannot be exploited by the MIP solver [24]. The computational results presented here demonstrate that valid inequalities from the ramping polytope may become more important as renewable resources come online causing large swings in net-load, but are not a restricting factor in our current ability to solve real-world unit commitment instances.
5 Conclusion

We have presented a compact extended formulation for a ramping-constrained generator and a cut-generating linear program based upon the extended formulation, which allows us to project separating cuts in to the typical 3-bin formulation for such generators. We demonstrated that these cuts are computational beneficial for high-wind unit-commitment instances based on the FERC generators and data from PJM. Finally, the slight relaxation of Balas's result [3, 4] presented in Section 3 may be of use in developing new extended formulations.

Acknowledgments

The authors are indebted to Stephen E. Wright for his useful comments in the preparation of this manuscript. This work is supported by NSF award number 1332662 and by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists, Office of Science Graduate Student Research (SCGSR) program. The SCGSR program is administered by the Oak Ridge Institute for Science and Education (ORISE) for the DOE. ORISE is managed by ORAU under contract number DE-AC05-06OR23100.

References


