The Ramping Polytope and Cut Generation 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

July 14, 2017

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 ramp up/down constraints, time-dependent start-up costs, and start-up/shut-down limits. 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 integer programming to optimize UC in place of Lagrangian relaxation (Carlson et al. 2012). Because of the scales involved, a 1% savings in energy markets results in translates to a $10 billion annual savings (O'Neill 2007).

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 feasible region. Frangioni and Gentile (2006) 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. During the drafting of this paper, we learned that Frangioni and Gentile independently discovered a similar perfect formulation for a single thermal generator (Frangioni and Gentile 2015a,b). The formulation developed can be used to model any properties of a generator that are polyhedrally representable when the commitment status is fixed. We use this extended formulation to create a cut-generating linear program that can be used to strengthen the linear programming (LP) relaxation for MIP formulations of UC and/or as a callback in the MIP solver.

The rest of the paper is outlined as follows. In Section 2 we review the current state of the unit commitment problem, including the typical 3-binary formulation used in state-of-the-art models, and introduce the extended formulation. In Appendix A, we prove the integrality of this extended formulation by revisiting a classic result of Balas (1979, 1998), and does so in more general terms than we strictly require. In Section 3 we use the results of the preceding sections to develop a cut-generating linear program for a ramping-constrained generator, and we present computational experiments based on a utility-scale unit commitment problem. Finally, in Section 4 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 $\mathcal{G}$ and T time steps, we formulate the unit commitment problem as follows:

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

$$\text{subject to} \sum_{g \in \mathcal{G}} p^g_t \geq L_t, \ \forall t \in [T] \tag{1b}$$

$$\sum_{g \in \mathcal{G}} \bar{p}^g_t \geq L_t + R_t, \ \forall t \in [T] \tag{1c}$$

$$p^g, \bar{p}^g \in \Pi^g \subset \mathbb{R}^T, \ \forall g \in \mathcal{G}, \tag{1d}$$

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. L_t is the electricity load at time t, while R_t is the spinning reserve requirement at time t. For convenience let $[T] := \{1, \ldots, T\}$. Π^g represents the often non-convex 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. The justification for this line of research is that tighter MIP formulations for Π^g increase the linear programming bound for (1), which will in turn decrease the enumeration necessary to solve UC.

Most of the literature modeling individual generators followed Garver's (1962) 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, and the last indicating if generator is turned off. Models of this type are referred to as 3-binary models, or *3-bin* models. An alternative 1-binary model (e.g., see Carrion and Arroyo 2006), 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 by Lee et al. (2004), 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 (Malkin 2003, Rajan and Takriti 2005).

The simplified generator considered by Rajan and Takriti (2005) 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 by Ostrowski et al. (2012); Damcı-Kurt et al. (2015) build on this by providing exponential classes of such strengthening inequalities along with a polynomial separation algorithm. Additionally, Damcı-Kurt et al. (2015) provide a convex hull description for the ramp-up and ramp-down polytopes in two time periods. Pan and Guan (2016) extend this by intersecting these two polytopes into an integrated ramping polytope for three time periods.

A convex hull description for the 3-bin model with the addition of start-up and shut-down power is proved by Gentile et al. (2016). The same authors extend this in Morales-España et al. (2015) by separating power from energy (often assumed to be the same since almost all UC models operate on a one-hour time interval). This allows for the modeling of a generator's output below economic minimum when starting up and shutting down, while still maintaining integrality.

Our model moves away from the standard 3-bin polytope by considering on-off intervals. To demonstrate the relationship between the proposed formulation and the classical formulations, we first review the standard 3-bin model typically used to represent Π^g and $c^g(\cdot)$. Then we turn our focus toward the dynamic programming method for optimizing over a single generator laid out by Frangioni and Gentile (2006). Indeed using their dynamic programming procedure 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 Π^g (although this is by no means guaranteed, see (Rothvoß 2014)). 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 Π^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 (Garver 1962):

$$u^g_t - u^g_{t-1} = v^g_t - w^g_t, \; \forall t \in [T], \tag{2}$$

and the minimum up/down time constraints (Rajan and Takriti 2005):

$$\sum_{i=t-UT+1}^{t} v_i^g \le u_t^g,\ \forall t \in [UT^g, T] \tag{3}$$

$$\sum_{i=t-DT+1}^{t} w_i^g \le 1 - u_t^g,\ \forall t \in [DT^g, T]. \tag{4}$$

Rajan and Takriti (2005) showed that (2 – 4) along with the variable bound constraints give a convex hull description for the minimum up/down time polytope.

Next we consider constraints on the generation limits. Let $\underline{P}$ and $\overline{P}$ represent the minimum and maximum feasible power output when on, RD and RU represent the maximum ramp-down and ramp-up 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

$$\underline{P}^g u_t^g \le p_t^g \le \bar{p}_t^g \le \overline{P}^g u_t^g,\ \forall t \in [T]. \tag{5}$$

We note if $RU^g, RD^g \ge (\overline{P}^g - \underline{P}^g)$ and $SU^g, SD^g \ge \overline{P}^g$, that is, there are no real ramping and start-up shut-down constraints, then the formulation given by (2 – 5) is perfect for this simple generator. However, most generators are not so simple, and have ramp-up constraints:

$$\bar{p}_t^g - p_{t-1}^g \le RU^g u_{t-1}^g + SU^g v_t^g,\ \forall t \in [T], \tag{6}$$

and ramp-down constraints:

$$\bar{p}_{t-1}^g - p_t^g \le RD^g u_t^g + SD^g w_t^g,\ \forall t \in [T]. \tag{7}$$

For reference later, define:

$$\Pi_{\text{3-bin}}^g := \{(p^g, \bar{p}^g, u^g, v^g, w^g) \in \mathbb{R}_+^{5T} \,|\, (2-7); (u^g, v^g, w^g) \in \{0,1\}^{3T}\} \tag{8}$$

and

$$^R\Pi_{\text{3-bin}}^g := \{(p^g, \bar{p}^g, u^g, v^g, w^g) \in \mathbb{R}_+^{5T} \,|\, (2-7); (u^g, v^g, w^g) \in [0,1]^{3T}\}. \tag{9}$$

That is, $\Pi_{\text{3-bin}}^g$ is the feasible set for the technical constraints for generator g, and $^R\Pi_{\text{3-bin}}^g$ is its continuous relaxation. We will colloquially refer to $\Pi_{\text{3-bin}}^g$ and $^R\Pi_{\text{3-bin}}^g$ as being in "3-bin space", dropping the g when it is implied by context.

Now we consider the cost function $c^g(\cdot)$. Typically $c^g(p^g) = c_f^g(u^g) + \sum_{t\in[T]} c_p^g(p_t^g)$, where $c_p^g(\cdot)$ is convex and either quadratic or piecewise linear in the power output, and $c_f(\cdot)$ is the fixed commitment costs and start-up/shut-down costs, and as such is a function of the indicator variables. First, we consider $c_p^g(\cdot)$. We assume that

$c_p^g(\cdot)$ is convex and piecewise linear where ${}^1\overline{P}^g, \ldots, {}^L\overline{P}^g$ represent the upper breakpoints for power available at marginal costs ${}^1c^g, \ldots, {}^Lc^g$ with ${}^1c^g < \ldots < {}^Lc^g$. Define ${}^0\overline{P}^g = 0$. We use the standard convex piecewise formulation by introducing new variables ${}^lp_t^g$, representing the power generator g produces at time t at marginal cost ${}^lc^g$, along with the constraints

$$0 \le {}^lp_t^g \le {}^l\overline{P}^g - {}^{l-1}\overline{P}^g, \ \forall l \in [L], \forall t \in [T] \tag{10a}$$

$$p_t^g = \sum_{l=1}^{L} {}^lp_t^g, \ \forall t \in [T]. \tag{10b}$$

We can then represent $c_p^g(\cdot)$ linearly as $\sum_{t\in[T]}\sum_{l\in[L]} {}^lc^g \, {}^lp_t^g$. Now consider $c_f^g(\cdot)$. 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 less than DT_C^g time periods. We formulate the start-up costs as in Morales-España et al. (2013); namely, let ${}^H\delta_t^g, {}^C\delta_t^g \in \{0, 1\}$ represent a hot and cold start-up, respectively. Then we may write ${}^H\delta_t^g, {}^C\delta_t^g$ in terms of the start-up and shut-down variables v_t^g, w_t^g:

$${}^H\delta_t^g \le \sum_{i=DT}^{DT_C^g - 1} w_{t-i}^g, \ \forall t \in [DT_C^g, T] \tag{11a}$$

$${}^H\delta_t^g + {}^C\delta_t^g = v_t^g, \ \forall t \in [T]. \tag{11b}$$

Thus, if ${}^Uc^g$ is the fixed cost of running the generator, and ${}^Dc^g$ is the cost of shutting down the generator, then we can represent $c_f^g(\cdot)$ linearly as $\sum_{t\in[T]}({}^Cc^g \, {}^C\delta_t^g + {}^Hc^g \, {}^H\delta_t^g + {}^Uc^g u_t^g + {}^Dc^g w_t^g)$.

If the ramping constraints (6) and (7) are irredundant, then it is well known that $\text{conv}(\Pi_{\text{3-bin}}^g) \neq {}^R\Pi_{\text{3-bin}}^g$ (where $\text{conv}(S)$ is the convex hull of the set S). Recently, Damcı-Kurt et al. (2015) characterized separately the ramp-up and ramp-down polytopes for when $T = 2$, and Pan and Guan (2016) fully characterized $\text{conv}(\Pi_{\text{3-bin}}^g)$ for $T = 3$. In the next section we will develop a new extended formulation for Π^g, which can be used to generate valid inequalities for $\text{conv}(\Pi_{\text{3-bin}}^g)$.

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 $D^{[a,b]} \subset \mathbb{R}^{2T}$ 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 $(p^{[a,b]}, \bar{p}^{[a,b]}) \in D^{[a,b]}$, $p_t^{[a,b]}$ represents the power produced by the generator at time t (note that $p_t^{[a,b]} = 0$ for all t not in the interval $[a, b]$), and $\bar{p}_t^{[a,b]}$ represents the maximum power available at time t. We can write $D^{[a,b]}$ as

$$D^{[a,b]} = \{(p^{[a,b]}, \bar{p}^{[a,b]}) \in \mathbb{R}_+^{2T} | A^{[a,b]}p^{[a,b]} + \bar{A}^{[a,b]}\bar{p}^{[a,b]} \le \bar{b}^{[a,b]}\} \tag{12}$$

for $A^{[a,b]}, \bar{A}^{[a,b]} \in \mathbb{R}^{m \times T}$ and $\bar{b}^{[a,b]} \in \mathbb{R}^m$. For our purposes, $D^{[a,b]}$ is defined by max/min power and ramping constraints, and is obviously bounded. The methods described in this paper then can be used to extend $D^{[a,b]}$ to accommodate any number of services so long as $D^{[a,b]}$ remains a bounded polyhedron.

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 $D^{[a,b]}_{typical}$ are:

$$p_t^{[a,b]} \leq 0 \qquad \forall t < a \text{ and } t > b \tag{13a}$$

$$\bar{p}_t^{[a,b]} \leq 0 \qquad \forall t < a \text{ and } t > b \tag{13b}$$

$$-p_t^{[a,b]} \leq -\underline{P} \qquad \forall t \in [a, b] \tag{13c}$$

$$p_t^{[a,b]} \leq \bar{p}_t^{[a,b]} \qquad \forall t \in [a, b] \tag{13d}$$

$$\bar{p}_t^{[a,b]} \leq \min(\overline{P}, SU + (t-a)RU, SD + (b-t)RD) \qquad \forall t \in [a, b] \tag{13e}$$

$$\bar{p}_t^{[a,b]} - p_{t-1}^{[a,b]} \leq \min(RU, SD + (b-t)RD - \underline{P}) \qquad \forall t \in [a+1, b] \tag{13f}$$

$$\bar{p}_{t-1}^{[a,b]} - p_t^{[a,b]} \leq \min(RD, SU + (t-a)RU - \underline{P}) \qquad \forall t \in [a+1, b]. \tag{13g}$$

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 the power output at time t. This ensures the generator does not produce more power than its maximum output $\overline{P}$, the power level it could ramp up to by time t ($SU + (t-a)RU$), or ramp down from at time t ($SD + (b-t)RD$) to reach shut-down status. The ramp up constraint (13f) ensures the power jump between times $t-1$ and t is no more than RU or that which we could ramp back down to in the remaining time ($SU + (b-t)RD - \underline{P}$). The ramp down constraints (13g) work symmetrically.

We will let $\mathcal{T}$ be the set of all feasible continuous operating intervals for the generator. Recalling UT and DT are the minimum up and down time for the generator, $\mathcal{T}$ contains all intervals $[a, b]$ where $1 \leq a \leq a + UT \leq b \leq T$. $\mathcal{T}$ also contains cases when the generator has been turned on prior to time one 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 $D^{[a,b]}_{typical}$ are nonempty for $[a, b] \in \mathcal{T}$. Frangioni and Gentile (2006) develop a dynamic-programming approach for scheduling a single generator in polynomial time by combining the polytopes $D^{[a,b]}_{typical}$ such that the intervals only overlap in feasible combinations. We will use the polytopes $D^{[a,b]}_{typical}$ 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 $\mathcal{T}$, 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 $V = \mathcal{T}$ 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 $\gamma \in \{0, 1\}^{|\mathcal{T}|}$ 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

$$\Gamma = \begin{cases} \sum_{\{[a,b]\in\mathcal{T} \mid t\in[a,b+DT]\}} \gamma_{[a,b]} \leq 1 & t \in [T] \\ \gamma_{[a,b]} \geq 0 & \forall [a,b] \in \mathcal{T} \end{cases} \tag{14}$$

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*

$$D := \begin{cases} A^{[a,b]}p^{[a,b]} + \bar{A}^{[a,b]}\bar{p}^{[a,b]} \leq \gamma_{[a,b]}\bar{b}^{[a,b]} & \forall [a,b] \in \mathcal{T} & \text{(15a)} \\ \sum_{[a,b]\in\mathcal{T}} p^{[a,b]} = p & & \text{(15b)} \\ \sum_{[a,b]\in\mathcal{T}} \bar{p}^{[a,b]} = \bar{p} & & \text{(15c)} \\ (p^{[a,b]}, \bar{p}^{[a,b]}) \in \mathbb{R}^{2T}_{+} & \forall [a,b] \in \mathcal{T} & \text{(15d)} \\ \sum_{\{[a,b]\in\mathcal{T} \mid t\in[a,b+DT]\}} \gamma_{[a,b]} \leq 1 & t \in [T] & \text{(15e)} \\ \gamma_{[a,b]} \geq 0 & \forall [a,b] \in \mathcal{T} & \text{(15f)} \end{cases}$$

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

Proof. Proof. See Appendix A. □

Remark 1. Not dispatching the generator in time period $[0, T + 1]$ corresponds to having $\gamma_{[a,b]} = 0$ for all $[a, b] \in \mathcal{T}$.

Linear generation costs $c \in \mathbb{R}^T$ and fixed start-up (and shut-down) costs $w \in \mathbb{R}^{|\mathcal{T}|}$ can be modeled by optimizing the linear function $c^\top p + w^\top \gamma$ over D. This formulation does not concern itself with time-dependent start-up costs. These can be easily added by considering additional indicator variables $\zeta_{[c,d]}$ which represent the generator being off from time c to d, and construct a set of feasible off intervals $\mathcal{T}'$ similar to the construction of $\mathcal{T}$. Recall ${}^{C}c$ and ${}^{H}c$ are the cost of a cold and hot start, respectively. When ${}^{H}c \geq {}^{C}c/2$, we can replace (15e) above with

$$\sum_{\{[a,b]\in\mathcal{T} \mid t\in[a,b+DT]\}} \gamma_{[a,b]} + \sum_{\{[c,d]\in\mathcal{T}' \mid t\in[c+DT,d]\}} \zeta_{[c,d]} = 1 \qquad t \in [T] \tag{16a}$$

$$\zeta_{[c,d]} \geq 0 \qquad \forall [c,d] \in \mathcal{T}', \tag{16b}$$

and the $\zeta_{[c,d]}$ variables are in the objective function with the appropriate objective value.

However, when ${}^{H}c < {}^{C}c/2$, a formulation discovered by Frangioni and Gentile (2015a,b) must be considered. It uses the following shortest path formulation in place of the packing formulation in (15e) and (15f) above

$$\sum_{\{[c,d]\in\mathcal{T}' \mid t=d+1\}} \zeta_{[c,d]} = \sum_{\{[a,b]\in\mathcal{T} \mid t=a\}} \gamma_{[a,b]} \qquad t \in [T] \tag{17a}$$

$$\sum_{\{[a,b]\in\mathcal{T} \mid t=b+1\}} \gamma_{[a,b]} = \sum_{\{[c,d]\in\mathcal{T}' \mid t=c\}} \zeta_{[c,d]} \qquad t \in [T] \tag{17b}$$

$$\sum_{\{[a,b]\in\mathcal{T} \mid a=0\}} \gamma_{[a,b]} + \sum_{\{[c,d]\in\mathcal{T}' \mid c=0\}} \zeta_{[c,d]} = 1 \tag{17c}$$

$$\sum_{\{[a,b]\in\mathcal{T} \mid b=T+1\}} \gamma_{[a,b]} + \sum_{\{[c,d]\in\mathcal{T}' \mid d=T+1\}} \zeta_{[c,d]} = 1 \tag{17d}$$

$$\gamma_{[a,b]} \geq 0,\ \forall [a,b] \in \mathcal{T},\ \zeta_{[c,d]} \geq 0,\ \forall [c,d] \in \mathcal{T}'. \tag{17e}$$

Call the resulting polytope D', that is, equations (15a - 15d) with (17). We note that Frangioni and Gentile (2015b) provide a proof of the integrality of D', building it up one dispatch polytope at a time, which also proves Theorem 1. It essentially relies on the fact that each of the combined polytopes are integer in their respective variables, and share only one variable with the other polytopes. In a similar fashion the results outlined in Appendix A prove the integrality of D' in the γ, ζ variables. Both proofs rely on the underlying integrality of the polytopes relating the indicator variables; the one presented in this paper also provides an interesting geometric interpretation of the result, whereas that in Frangioni and Gentile (2015b) is a bit more generalizable.

Both convex hull descriptions are large, and are unlikely to be computationally effective within the problem (1). Preliminary computational experiments in Frangioni and Gentile (2015b) bear this out. As such, instead of using (15) directly to represent Π^g in the unit commitment problem (1), we will use (15) to develop a procedure for generating cuts based on the polytope D similar in spirit to lift-and-project cuts (Balas et al. 1993).

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

The basic logic of the proposed approach is as follows. Let D^g be the feasible dispatch polytope for generator g. The LP relaxation for (1) when the extended formulation is used to represent each generator is

$$^{LP}x^*_{EF} = \min \sum_{g \in \mathcal{G}} c^g(p^g) \tag{18a}$$

$$\text{subject to} \sum_{g \in \mathcal{G}} p^g_t \geq L_t, \ \forall t \in [T] \tag{18b}$$

$$\sum_{g \in \mathcal{G}} \bar{p}^g_t \geq L_t + R_t, \ \forall t \in [T] \tag{18c}$$

$$(p^g, \bar{p}^g) \in D^g, \ \forall g \in \mathcal{G}. \tag{18d}$$

On the other hand, define the LP relaxation for (1) with the typical 3-bin formulation as

$$^{LP}x^*_{\text{3-bin}} = \min \sum_{g \in \mathcal{G}} c^g(p^g) \tag{19a}$$

$$\text{subject to} \sum_{g \in \mathcal{G}} p^g_t \geq L_t, \ \forall t \in [T] \tag{19b}$$

$$\sum_{g \in \mathcal{G}} \bar{p}^g_t \geq L_t + R_t, \ \forall t \in [T] \tag{19c}$$

$$(p^g, \bar{p}^g) \in {}^R\Pi^g_{\text{3-bin}}, \ \forall g \in \mathcal{G}. \tag{19d}$$

Because the extended formulation is at least as tight as 3-bin, we have $^{LP}x^*_{EF} \geq {}^{LP}x^*_{\text{3-bin}}$. However, as mentioned above, because representing D^g requires $\mathcal{O}(T^3)$ variables whereas $^R\Pi^g_{\text{3-bin}}$ needs only $\mathcal{O}(T)$ variables, the problem (18) is likely to be much more computationally difficult than (19). In the MIP context, this not only slows down the root-node solve time, but also subsequent node resolves in the branch-and-bound tree.

We propose a cutting-plane procedure attempts to ameliorate the computational issues of solving (18) while still maintaining the strength of its LP bound. In particular, given a solution to (19), for each $g \in \mathcal{G}$ we can lift generator schedules $(p^{g*}, \bar{p}^{g*}) \in {}^R\Pi^g_{\text{3-bin}}$ to the "D^g-space." If $(p^{g*}, \bar{p}^{g*}) \in D^g$, then we do nothing. On the other hand, if $(p^{g*}, \bar{p}^{g*}) \notin D^g$, we can calculate a separating cut, project the cut back into 3-bin space, add it to the problem (19), and then resolve. Repeating this process iteratively until $(p^{g*}, \bar{p}^{g*}) \in D^g$ for all $g \in \mathcal{G}$ allows us to calculate $^{LP}x^*_{EF}$ while decomposing the difficult constraints D^g into individual easier separation subproblems for each $g \in \mathcal{G}$. Further, we can stop at any point and possibly obtain a better LP bound than ${}^{LP}x^*_{\text{3-bin}}$. That is, for each $g \in \mathcal{G}$ let C^g be the feasible region for a (possibly empty) set of cuts generated from D^g. Consider the following linear program:

$$
\begin{aligned}
{}^{LP}x^*_{\text{3-bin}+C} = \min \; & \sum_{g \in \mathcal{G}} c^g(p^g) && \text{(20a)}\\
\text{subject to} \; & \sum_{g \in \mathcal{G}} p^g_t \geq L_t, \; \forall t \in [T] && \text{(20b)}\\
& \sum_{g \in \mathcal{G}} \bar{p}^g_t \geq L_t + R_t, \; \forall t \in [T] && \text{(20c)}\\
& (p^g, \bar{p}^g) \in {}^{R}\Pi^g_{\text{3-bin}} \cap C^g, \; \forall g \in \mathcal{G}. && \text{(20d)}
\end{aligned}
$$

Then we have ${}^{LP}x^*_{EF} \geq {}^{LP}x^*_{\text{3-bin}+C} \geq {}^{LP}x^*_{\text{3-bin}}$, where we achieve equality on the left if we add every possible separating cut, and we have equality on the right if we add no cuts. As is well known, in practice is it often not desirable to add every possible cut, so we add cuts heuristically (in this work, we do only one round of cuts for a given LP relaxation). Finally, note that the discussion above not only applies to the root node of the branch-and-bound tree, but also at any subsequent node subproblem.

3.1 From Dispatch Polytope Space to 3-bin Space

Recalling 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.

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 γ and (u, v, w) as follows:

$$
\begin{aligned}
\sum_{\{[a,b] \in \mathcal{T} \;|\; t \in [a,b]\}} \gamma_{[a,b]} &= u_t && t \in [T] && \text{(21a)}\\
\sum_{\{[a,b] \in \mathcal{T} \;|\; t = a\}} \gamma_{[a,b]} &= v_t && t \in [T] && \text{(21b)}\\
\sum_{\{[a,b] \in \mathcal{T} \;|\; t = b+1\}} \gamma_{[a,b]} &= w_t && t \in [T]. && \text{(21c)}
\end{aligned}
$$

Notice by adding the constraints (21) to the formulation of D (15), for a given 3-bin solution $(p^*, \bar{p}^*, u^*, v^*, w^*) \in {}^{R}\Pi_{\text{3-bin}}$, either the system of equations defined by (15), (21), $p = p^*$, $\bar{p} = \bar{p}^*$, $u = u^*$, $v = v^*$, and $w = w^*$, will be feasible, in which case this 3-bin solution is in the ramping polytope, or this system of equations 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 Farkas certificate for the system of equations to generate a cut for the 3-bin space which cuts-off this infeasible solution. Our cut-generating linear program picks, in some sense, the best such infeasibility certificate so as to get the deepest cut.

3.2 A Cut-Generating Linear Program

For ease we will consider the dual form of the cut-generating LP, which is derived from (15) and (21) above. Let e be the appropriately sized vector of 1's and suppose $z \in \mathbb{R}_+$. Let $(p^*, \bar{p}^*, u^*, v^*, w^*)$ be a solution vector in 3-bin space. Consider the following linear program:

$$z^* = \min z \tag{22a}$$

subject to

$$\begin{aligned}
&\pi && A^{[a,b]}p^{[a,b]} + \bar{A}^{[a,b]}\bar{p}^{[a,b]} \le \gamma_{[a,b]}\bar{b}^{[a,b]} + ze && \forall [a,b] \in \mathcal{T} && \text{(22b)}\\
&\delta && \sum_{\{[a,b]\in\mathcal{T}\ \mid\ t\in[a,b+DT]\}} \gamma_{[a,b]} \le 1 + z && t \in [T] && \text{(22c)}\\
&\varepsilon && \sum_{[a,b]\in\mathcal{T}} p^{[a,b]} = p^* && && \text{(22d)}\\
&\mu && \sum_{[a,b]\in\mathcal{T}} \bar{p}^{[a,b]} = \bar{p}^* && && \text{(22e)}\\
&\xi && \sum_{\{[a,b]\in\mathcal{T}\ \mid\ t\in[a,b]\}} \gamma_{[a,b]} = u_t^* && t \in [T] && \text{(22f)}\\
&\alpha && \sum_{\{[a,b]\in\mathcal{T}\ \mid\ t=a\}} \gamma_{[a,b]} = v_t^* && t \in [T] && \text{(22g)}\\
&\sigma && \sum_{\{[a,b]\in\mathcal{T}\ \mid\ t=b+1\}} \gamma_{[a,b]} = w_t^* && t \in [T] && \text{(22h)}\\
& && z \in \mathbb{R}_+;\ \ p^{[a,b]}, \bar{p}^{[a,b]} \in \mathbb{R}^T_+, \gamma_{[a,b]} \in \mathbb{R}_+, && \forall [a,b] \in \mathcal{T}, && \text{(22i)}
\end{aligned}$$

where $\pi \in \mathbb{R}_-^{m|\mathcal{T}|}$ is the set of dual variables for constraints (22b), $\delta \in \mathbb{R}^T_-$ is the set of dual variables for (22c), and $\varepsilon, \mu, \xi, \alpha, \sigma \in \mathbb{R}^T$ are the sets of dual variables for constraints (22d - 22h), respectively. We observe if z^* is 0, then $(p^*, \bar{p}^*)$ is a feasible solution to D, and if not, we can use the optimal dual vector to cut off the 3-bin solution $(p^*, \bar{p}^*, u^*, v^*, w^*) \in {}^R\Pi_{\text{3-bin}}$. To demonstrate, suppose $z^* > 0$ and we have an optimal dual vector $\pi^*, \delta^*, \varepsilon^*, \mu^*, \xi^*, \alpha^*, \sigma^*$. Then by strong duality $z^* = (\delta^*)^T e + (\varepsilon^*)^T p^* + (\mu^*)^T \bar{p}^* + (\xi^*)^T u^* + (\alpha^*)^T v^* + (\sigma^*)^T w^* > 0$, and so the cut $(\delta^*)^T e + (\varepsilon^*)^T p + (\mu^*)^T \bar{p} + (\xi^*)^T u + (\alpha^*)^T v + (\sigma^*)^T w \le 0$ 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 $\text{conv}(\Pi_{\text{3-bin}})$).

We maximize the depth of the cut by choosing the optimal such cut, with respect to the 1-norm normalization, instead of any dual feasible solution to (22) (Balas et al. 1993). In particular, note in the dual of (22), the constraint associated with z is $-e^T\delta - e^T\pi \leq 1$. When the proposed solution $(p^*, \bar{p}^*, u^*, v^*, w^*)$ is infeasible for ${}^R\Pi_{\text{3-bin}}$, this limits the 1-norm of these otherwise unbounded rays.

3.3 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. These consist of two sets of generators, a "summer" and "winter" set of generators, which are based on market data provided from the PJM Interconnection and other sources (Krall et al. 2012). We use the standard 3-bin formulation for the master unit commitment MIP, as discussed in Section 2.1, that is:

$$\min \sum_{g\in\mathcal{G}} \sum_{t\in[T]} \left(\sum_{l\in[L]} ({}^l c^g\, {}^l p_t^g) + {}^C c^g\, {}^C\delta_t^g + {}^H c^g\, {}^H\delta_t^g + {}^R c^g u_t^g + {}^D c^g w_t^g \right) \tag{23a}$$

$$\text{s.t.} \sum_{g\in\mathcal{G}} p_t^g \geq L_t,\ \forall t \in [T] \tag{23b}$$

$$\sum_{g\in\mathcal{G}} \bar{p}_t^g \geq L_t + R_t,\ \forall t \in [T] \tag{23c}$$

$$(10), (11)\ \forall g \in \mathcal{G}, \tag{23d}$$

$$(p^g, \bar{p}^g, u^g, v^g, w^g) \in \Pi_{\text{3bin}}^g,\ \forall g \in \mathcal{G}. \tag{23e}$$

We only consider cuts on a subset of the generators, namely for those that have irredundant ramping constraints while operating and those that have a minimum run time of at least 2. That is, we consider cuts on $\mathcal{G}^C := \{g \in \mathcal{G} \mid (\overline{P}^g - \underline{P}^g) > \min\{RD^g, RU^g\}$ and $UT^g \geq 2\}$. We do this because the 3-bin formulation is tight for generators with no ramping constraints and the problem (22) becomes impractically large for generators with $UT^g = 1$.

The cuts are implemented in a callback, namely, given the current LP relaxation for (23), for each generator in $\mathcal{G}^C$ with fractional status variables, we use (22) to determine if $(p^*, \bar{p}^*, u^*, v^*, w^*) \in \text{conv}(\Pi_{\text{3-bin}})$, and if not we add the violated ramping inequality given by the optimal dual solution of (22). We generate the cuts using the "bundling" approach of Balas et al. (1993), that is, at the current master LP relaxation we try to generate a cut for each $g \in \mathcal{G}^C$ and give these to the solver together. This is visualized in Figure 1, where we see for each

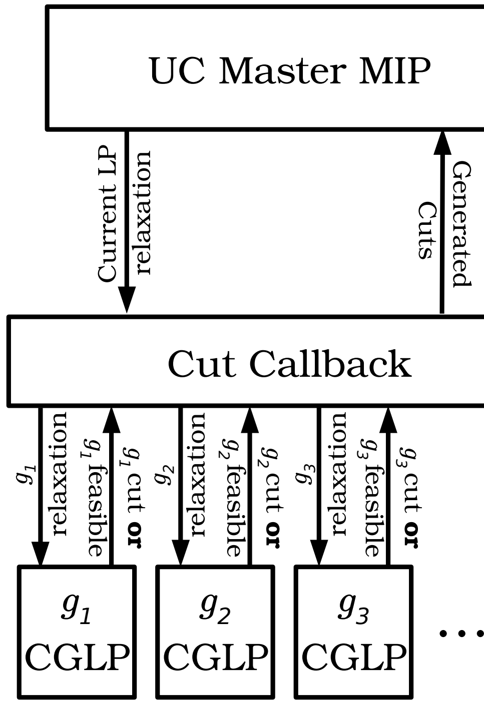

Figure 1: Visualization of the cut-generation procedure

unit $g \in \mathcal{G}^C$ the cut-generation LP (CGLP) is an independent problem. For a given LP relaxation to the UC Master MIP (23), we separate the solution by generator g, and then check the feasibility of this relaxed solution in D^g by solving the CGLP (22). If we find the relaxed solution infeasible, we pass the generated cut to the UC Master MIP, along with the cuts generated for other units in $\mathcal{G}^C$. Though we do not do so for this paper, it would be trivial to parallelize the cut-generation procedure.

Next, we discuss some computational enhancements to this general outline. First, to mitigate numerical issues, we only add cuts for which the solution to (22), z^*, is greater than 10^{-2}. Second, after the first 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. Third, we make an enhancement based on symmetry by observing that if $g_1, g_2 \in \mathcal{G}^C$ have identical parameters (including initial conditions), 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.

Table 1: 3-bin UC Formulation Problem Size.

	# cont. vars	# binary vars	# constraints	# nonzeros
Winter System	102048	112220	288478	963303
Summer System	103584	117360	300108	983799

While this is heavy machinery for easy UC instances, we will see that for hard UC instances having this machinery available results in a noticeable improvement.

3.4 Computational Experiments

All computational experiments were performed on a Dell PowerEdge T620 with 2 Intel Xeon E5-2670 processors and 256GB of RAM 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 (PJM 2016a,b). For each day in 2015 a 24-hour unit commitment problem was formulated, with wind generation accounted for as negative demand in (23). 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 ramp rates are provided, $SU^g = SD^g = \underline{P}^g$ 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.

In Table 1 we specify the size of the base UC formulation (23) for both the winter and summer set of generators, reporting the number of constraints, continuous variables, binary variables, and non-zero elements

Table 2: Winter System: Cut Generation LP Sizes.

	# variables	# constraints	# nonzeros
Mean	7382.1	12247.2	46819.3
Min	1177	2148	7788
Median	8674	14251	53971
Max	14701	22572	84701

Table 3: Summer System: Cut Generation LP Sizes.

	# variables	# constraints	# nonzeros
Mean	7568.3	12537.3	47918.5
Min	1177	2148	7788
Median	8674	14778.5	56977
Max	14701	22572	84701

in the constraints matrix, respectively. These form the template for our test set, as we vary only the demand and reserve data. As we can see, both generator sets yield MIPs of significant size.

In Table 2 we report some summary statistics for the 459 cut generation LPs for the winter system, and similarly in Table 3 we report summary statistics for the 492 cut generation LPs for the summer system, based on the formulation (22). As we can see, most of the LPs are of modest size, with the variation dependent on the parameters UT^g and DT^g. Namely, the larger UT^g and DT^g are, the fewer valid time intervals in $\mathcal{T}$ for generator g, so the smaller the formulation (22) is. As an example, many of the generators in the test sets are large coal units with $UT^g = 15$ and $DT^g = 9$. These units only have 2059 variables in their cut generation LPs. Some of the nuclear units in the test sets only have 1177 variables in their cut generation LPs because once started (stopped) they must stay on (off) for the rest of the time horizon. Conversely, there are some small gas units in both test sets which have $UT^g = DT^g = 2$. These small units have larger cut generation LPs with 14701 variables.

3.5 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 4. 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,

Table 4: Initial Computational Results.

	No User Cuts		Ramping Polytope Cuts			
	Time (s)	Nodes	Time (s)	Nodes	Cuts Added	Cut Time (s)
Geometric Mean	172.52	44.01	163.39	44.08	97.16	10.50
Min	85.48	0	82.13	0	16	1.24
Median	168.97	0	153.82	0	99	11.83
Max	557.82	715	812.32	640	636	77.13
# inst. better	130	37	231	26		

Table 5: High Wind Computational Summary, Solved Instances.

	No User Cuts		Ramping Polytope Cuts			
	Time (s)	Nodes	Time (s)	Nodes	Cuts Added	Cut Time (s)
Geometric Mean	204.47	60.84	196.50	48.03	153.99	8.34
Min	58.40	0	54.78	0	2	0.19
Median	197.41	0	192.63	0	173	10.32
Max	1523.65	9875	1030.74	10976	804	58.50
# inst. better	138	57	218	73		

most instances are solved at the root node, and no instance takes more than about 15 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 last row we report the number of instances for which that method is strictly better. Here we see the cuts usually result in a better time, though just slightly, and in most cases ($n = 298$) both methods need the same number of nodes to prove optimality to the default tolerance of 0.01% (as most are solved at the root).

3.6 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 (GE Energy 2014). 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

Table 6: High Wind, Harder Instances.

	No User Cuts		Ramping Polytope Cuts			
Date	Time (s)	Nodes	Time (s)	Nodes	Cuts Added	Cut Time (s)
04 Jan	601.93	2841	543.98	1935	208	7.58
15 Mar	988.85	4129	751.59	1533	222	8.77
01 Apr	704.16	554	458.90	216	525	21.18
03 Apr	644.06	689	540.69	544	367	13.84
12 Apr	742.06	1310	435.95	556	125	13.45
15 Apr	1523.65	627	615.81	374	742	26.00
25 Apr	1152.09	1440	1030.74	1220	413	15.89
24 May	988.10	686	377.98	162	505	14.19
23 Jun	723.91	583	447.34	575	298	40.49
02 Oct	620.03	5030	911.74	10976	108	7.65
24 Oct	556.26	934	608.30	772	21	7.49
28 Oct	411.65	1134	701.46	3764	133	6.59
21 Nov	1193.51	4125	940.53	3130	187	9.30
25 Nov	567.15	2648	810.62	3910	288	13.08
16 Dec	993.68	5946	758.29	3214	192	8.54
23 Dec	793.52	9875	570.36	3688	94	6.47
Geometric Mean	780.74	1742.70	629.51	1255.49	210.66	11.88

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 true.

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 5. As we can see, there are modest reductions in geometric mean solve time and geometric mean nodes. 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 6. 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 geometric mean reduction in run time of about 150 seconds, such that the typical hard instance went from

Table 7: High Wind, Timed Out Instances.

	No User Cuts		Ramping Polytope Cuts			
Date	MIP Gap	Nodes	MIP Gap	Nodes	Cuts Added	Cut Time (s)
27 Oct	0.0102%	10202	0.0116%	8024	400	15.69
12 Nov	0.1287%	2613	0.0871%	3470	447	13.50
14 Nov	0.0111%	10202	0.0104%	8939	561	17.88
17 Nov	0.1015%	2242	0.1144%	1917	729	13.29
26 Nov	0.2110%	4778	0.2128%	4498	225	9.83
20 Dec	0.0232%	10202	0.0188%	9955	243	12.49

taking approximately 13 minutes to 10.5 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 28% reduction in geometric 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 7 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.

3.7 Observed Cuts

To better understand the cuts generated from (22), we examined the generated cuts for a subset of the high-wind test instances. The vast majority of the cuts were variable upper-bound inequalities. Specifically, those of the form

$$\bar{p}_t^g \leq \sum_{i\in[T]} (\xi_i^g u_i^g + \alpha_i^g v_i^g + \sigma_i^g w_i^g)\,, \tag{24}$$

for some $t \in [T]$, where the coefficients ξ_i^g, α_i^g, and σ_i^g are the normalized optimal dual values from (22) (so that the coefficient on $\bar{p}_t^g$ is 1).

In a similar fashion two-period ramping inequalities were observed, i.e.

$$\bar{p}_t^g - p_{t-j}^g \leq \sum_{i\in[T]} (\xi_i^g u_i^g + \alpha_i^g v_i^g + \sigma_i^g w_i^g)\,, \tag{25}$$

$$\bar{p}_{t-j}^g - p_t^g \leq \sum_{i\in[T]} (\xi_i^g u_i^g + \alpha_i^g v_i^g + \sigma_i^g w_i^g)\,, \tag{26}$$

where j was most often 1, but sometimes 2, and on one occasion 3 for the ramp-up inequality (25). Three-period ramping inequalities were also common

$$-p^g_{t-j} + \bar{p}^g_t - p^g_{t+k} \le \sum_{i\in[T]} \left(\xi^g_i u^g_i + \alpha^g_i v^g_i + \sigma^g_i w^g_i\right), \tag{27}$$

$$\bar{p}^g_{t-j} - p^g_t + \bar{p}^g_{t+k} \le \sum_{i\in[T]} \left(\xi^g_i u^g_i + \alpha^g_i v^g_i + \sigma^g_i w^g_i\right), \tag{28}$$

with (28) occurring much more often than (27), and both usually having $j = k = 1$. A few inequalities of the form (28) were observed with $j = 1$, $k = 2$ and $j = 2$, $k = 1$, and at least one instance with $j = 1$, $k = 3$.

Occasionally more exotic inequalities would be generated. The four-period ramping inequalities

$$-p^g_{t-j} + \bar{p}^g_t - p^g_{t+k} + \bar{p}^g_{t+l} \le \sum_{i\in[T]} \left(\xi^g_i u^g_i + \alpha^g_i v^g_i + \sigma^g_i w^g_i\right), \tag{29}$$

$$\bar{p}^g_{t-j} - p^g_t + \bar{p}^g_{t+k} - p^g_{t+l} \le \sum_{i\in[T]} \left(\xi^g_i u^g_i + \alpha^g_i v^g_i + \sigma^g_i w^g_i\right), \tag{30}$$

with consecutive time periods ($j = 1$, $k = 1$, $l = 2$) were most common, but four-period inequalities with $j = 2$, $k = 1$, and $l = 2$ were observed as well. The five-period ramping inequality

$$\bar{p}^g_{t-2} - p^g_{t-1} + \bar{p}^g_t - p^g_{t+1} + \bar{p}^g_{t+2} \le \sum_{i\in[T]} \left(\xi^g_i u^g_i + \alpha^g_i v^g_i + \sigma^g_i w^g_i\right), \tag{31}$$

was also generated on several occasions. In a similar vain, a seven-, nine-, and ten-period ramping inequalities were observed once.

In all cases the generated inequalities had varying degrees of sparsity in the generator's status variables (u^g, v^g, w^g). Some cuts were generated with only two non-zeros on the right-hand side, these were always involving the end of the time horizon. Several inequalities only had a few non-zeros in the right-hand side. Many more however spanned the generator's production horizon (i.e., when the u^g variables are non-zero), though in most cases these inequalities had about one-third to one-half non-zeros on the right-hand side. This is because with fractional status variables, the cut generated usually spans several of the polytopes $D^{[a,b]}$.

3.8 Reflections

First, we note that it is somewhat surprising that we were able to separate so many cuts in a reasonable amount of time. However, with $T = 24$ we see from Tables 2 and 3 that many of the LPs are of a manageable size, and most can be solved quickly with a modern commercial LP solver. Further, after the first cut pass the most of the generators have the same solution in the UC LP relaxation, so there's nothing for Gurobi to do in the LP separation problem. (As mentioned above, we leave the LP separation problems loaded in memory.)

Additionally, across all the test instances, 33% of the cuts we add are additional valid cuts added by symmetry. (Recall that if we compute a cut for a generator we also add that cut for all generators in its orbit.)

Even though in practice we were able to solve the cut-generating problem for $T = 24$, obviously the problem (22) increases super-linearly in T. One way to improve the performance could be to more carefully screen which generators we generate cuts on. It may only make sense with longer time horizons to generate cuts on those "large" generators (with large UT^g and DT^g) for which the super-linear explosion in formulation size is more manageable. Additionally, it is clear from the formulation of (22) that most of the columns and rows (specifically those from (22b)) probably never enter the basis, and could perhaps be generated on the fly, with an initial basis constructed based on the 3-bin solution. However, the implementation of such a method would be non-trivial. Another possibility to improve the performance on longer time horizons would be to solve a "rolling" separation problem, where for each time period t we solve a small (e.g. 7-period) version of (22) centered around time t. That being said, the density of the cuts observed suggests that such a procedure may be less effective at generating quality cuts. Alternatively, problem (22) could be generated on the fly based on the LP relaxation values for u^g. Notice (22) decomposes when $u_t^* = 0$ for some t, and though adding time-dependent start-up costs complicates this picture, such a decomposition could be done heuristically.

Finally, 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 Damcı-Kurt et al. (2015) and Pan and Guan (2016). We note that the cuts given by (22) are a superset of those presented in these two papers. The "slow-start" generators in Damcı-Kurt et al. (2015) take an average of 4 time periods to ramp from SU to $\overline{P}$, and the "fast-start" generators need an average of 3 time periods. Similarly, for the instances used in Pan and Guan (2016), every generator in the test set needs 4 time periods to ramp up to $\overline{P}$. 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 (Ostrowski et al. 2015). In the systems we test here, 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 $\overline{P}$, and half (which we do not generate inequalities for) do not have ramping constraints at all. As far as the authors are aware this is the first paper to test any ramping inequalities based on real-world generator data. The computational results presented here demonstrate that valid inequalities from the ramping polytope are beneficial for difficult unit commitment instances, and do not detract from unit commitment instances which are easy to solve.

4 Conclusion

We have presented a compact extended formulation for a ramping-constrained generator and a cut-generating linear program based upon the extended formulation. We demonstrated that the these cuts are computational beneficial for high-wind unit commitment instances based on the FERC generator set and data from PJM. Finally, the slight generalization of Balas's result (Balas 1979, 1998) presented in Appendix A 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 Educataion (ORISE) for the DOE. ORISE is managed by ORAU under contract number DE-AC05-06OR23100.

References

Balas E (1979) Disjunctive programming. *Annals of Discrete Mathematics* 5:3–51.

Balas E (1998) Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics* 89(1):3–44.

Balas E, Ceria S, Cornuéjols G (1993) A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming* 58(1-3):295–324.

Bonami P, Lodi A, Tramontani A, Wiese S (2015) On mathematical programming with indicator constraints. *Mathematical Programming* 151(1):191–223.

Carlson B, Chen Y, Hong M, Jones R, Larson K, Ma X, Nieuwesteeg P, Song H, Sperry K, Tackett M, et al. (2012) MISO unlocks billions in savings through the application of operations research for energy and ancillary services markets. *Interfaces* 42(1):58–73.

Carrion M, Arroyo JM (2006) A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems* 21(3):1371–1378.

Conforti M, Cornuéjols G, Zambelli G (2014) *Integer Programming*, volume 271 (Springer).

Damcı-Kurt P, Küçükyavuz S, Rajan D, Atamtürk A (2015) A polyhedral study of production ramping. *Mathematical Programming* 1–31.

Faenza Y, Oriolo G, Stauffer G (2010) The hidden matching structure of the composition of strips: a polyhedral perspective. *14th Aussois Workshop on Combinatorial Optimization, Aussois (January 2010)*.

Frangioni A, Gentile C (2006) Solving nonlinear single-unit commitment problems with ramping constraints. *Operations Research* 54(4):767–775.

Frangioni A, Gentile C (2015a) An extended MIP formulation for the single-unit commitment problem with ramping constraints. *17th British-French-German conference on Optimization* (London).

Frangioni A, Gentile C (2015b) New MIP formulations for the single-unit commitment problems with ramping constraints. IASI Research Report 15-06.

Garver LL (1962) Power generation scheduling by integer programming-development of theory. *Power Apparatus and Systems, Part III. Transactions of the American Institute of Electrical Engineers* 81(3):730–734.

GE Energy (2014) PJM renewable integration study. PJM Interconnection.

Gentile C, Morales-Espana G, Ramos A (2016) A tight mip formulation of the unit commitment problem with start-up and shut-down constraints. *EURO Journal on Computational Optimization* 1–25.

Grünbaum B (2003) *Convex polytopes, volume 221 of Graduate Texts in Mathematics* (Springer-Verlag).

Jeroslow RG (1987) Representability in mixed integer programming, I: characterization results. *Discrete Applied Mathematics* 17(3):223–243.

Krall E, Higgins M, O'Neill RP (2012) RTO unit commitment test system. Federal Energy Regulatory Commission.

Lee J, Leung J, Margot F (2004) *Discrete Optimization* 1(1):77–85.

Magnanti TL, Wolsey LA (1995) Optimal trees. *Handbooks in Operations Research and Management Science* 7:503–615.

Malkin P (2003) Minimum runtime and stoptime polyhedra. CORE Report, Université catholique de Louvain.

Morales-España G, Gentile C, Ramos A (2015) Tight MIP formulations of the power-based unit commitment problem. *OR Spectrum* 1–22.

Morales-España G, Latorre JM, Ramos A (2013) Tight and compact MILP formulation of start-up and shut-down ramping in unit commitment. *IEEE Transactions on Power Systems* 28(2):1288–1296.

O'Neill RP (2007) It's getting better all the time (with mixed integer programming). HEPG Forty-Ninth Plenary Session.

Ostrowski J, Anjos MF, Vannelli A (2012) Tight mixed integer linear programming formulations for the unit commitment problem. *IEEE Transactions on Power Systems* 27(1):39.

Ostrowski J, Anjos MF, Vannelli A (2015) Modified orbital branching for structured symmetry with an application to unit commitment. *Mathematical Programming* 150(1):99–129.

Pan K, Guan Y (2016) A polyhedral study of the integrated minimum-up/-down time and ramping polytope. arXiv:1604.02184.

PJM (2016a) PJM - ancillary services. http://pjm.com/markets-and-operations/ancillary-services.aspx, accessed: 2016-01-07.

PJM (2016b) PJM - system operations. http://www.pjm.com/markets-and-operations/ops-analysis.aspx, accessed: 2016-01-07.

Pochet Y, Wolsey LA (1993) Lot-sizing with constant batches: Formulation and valid inequalities. *Mathematics of Operations Research* 18(4):767–785.

Rajan D, Takriti S (2005) Minimum up/down polytopes of the unit commitment problem with start-up costs. IBM Research Report, RC23628 (W0506-050).

Rockafellar R (1970) *Convex Analysis* (Princeton University Press).

Rothvoß T (2014) The matching polytope has exponential extension complexity. *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, 263–272 (ACM).

Stoer J, Witzgall C (1970) *Convexity and Optimization in Finite Dimensions* (Springer).

Weyl H (1950) The elementary theory of convex polyhedra. *Contributions to the Theory of Games* 1(24):3–18.

A Constrained Minkowski Sums of Polyhedra

We will prove Theorem 1 by extending the classical result of Balas on disjunctive programs. The success of disjunctive programming as initially laid out by Balas (1979, 1998) toward the practical solvability of problems involving indicator constraints is clear; see (Bonami et al. 2015) for a recent overview. We consider an extension of Balas's classical result (Theorem 2), a weaker version of which is given as a lemma by Faenza et al. (2010), and show it can be used to model constrained Minkowski sums of polyhedra. Generalizing the convexity constraint is an approach that has been taken before, namely for the shortest path polytope (Pochet and Wolsey 1993) and the tree packing polytope (Magnanti and Wolsey 1995), which would be sufficient to prove Theorem 1. However, we show that any integer polytope could be used in place of the convexity constraint, allowing for a great deal of modeling flexibility. 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

$$\lambda C := \{\lambda x \mid x \in C\}, \tag{32}$$

$$C_1 + C_2 := \{x_1 + x_2 \mid x_1 \in C_1,\ x_2 \in C_2\}. \tag{33}$$

For a set $S \subset \mathbb{R}^n$, conv(S) is the convex hull of S and 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 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 (Rockafellar 1970, Stoer and Witzgall 1970, Grünbaum 2003), with the Minkowski-Weyl theorem for polyhedra (Weyl 1950) 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 conv $\left(\cup_{i\in[m]} P^i\right) = \bigcup\{\sum_{i=1}^m \gamma_i P^i \mid \sum_{i=1}^m \gamma_i = 1, \gamma \geq 0\}$. An interesting question is when is such a set closed and polyhedral. Indeed Theorem 9.8 and subsequent corollaries in Rockafellar (1970) give sufficient conditions for closedness. Balas (1979, 1998) provides sufficient conditions for polyhedreality 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 $S =$ conv $\left(\cup_{i\in[m]} P^i\right)$ and polyhedron $P =$ conv$(\cup_{i\in[m]} Q^i) +$ cone$(\cup_{i\in[m]} R^i)$. Then the polyhedron*

$$Y = \begin{cases} A^i x^i \leq \gamma_i b^i,\ i \in [m] \\ \sum_{i\in[m]} x^i = x \\ \sum_{i\in[m]} \gamma_i = 1 \\ \gamma_i \geq 0,\ i \in [m] \end{cases} \tag{34}$$

provides an extended formulation of P. If each P^i, $i \in [m]$, is nonempty then cl$(S) = P$. Additionally, the vertices of Y have binary γ_i.

In the context of Theorem 2 we also have the following result from Jeroslow (1987) and Corollary 9.8.1 in Rockafellar (1970):

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 Γ 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 those of the preceding theorems? We answer this question in the affirmative, with a few restrictions on Γ.

To see what some of these restrictions must be, consider the challenges of using indicator variables as in (34). 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 γ 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 γP when γ 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 (34), while the indicator variables γ 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 (34). 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 A.1 to include possibly empty polyhedra.

A.1 The Extended Formulation

Now consider the set $S := \bigcup_{\gamma \in \Gamma} \left(\sum_{i=1}^m \gamma_i P^i\right)$, 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 Conforti et al. 2014, Section 4.9.

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*

$$Y := \begin{cases} A^i x^i \leq \gamma_i b^i, \ i \in [m] \\ \sum_{i=1}^m x^i = x \\ (\gamma_1, \ldots, \gamma_m) = \gamma \in \Gamma. \end{cases} \tag{35}$$

Then $P = proj_x(Y) := \{x \in \mathbb{R}^n \mid \exists(x^1, \ldots, x^m, \gamma) \in \mathbb{R}^{nm+m}$ 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 $A^i x^i = A^i(\gamma_i q^i + r^i) = \gamma_i A^i q^i + A^i r^i \leq \gamma_i b^i + 0$ 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. $\square$

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.

The next theorem demonstrates that $\text{cl}(S) = P$ with a restriction on Γ.

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

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 $x \in P$.

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

The requirement that Γ 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 P^i since it never contributes to the sum.

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

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

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

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

It may be that Γ is the continuous relaxation of some integer set which determines the polyhedra P^i simultaneously allowed in the sum. The next theorem shows that vertices and extreme rays of Y have γ components which are vertices and extreme rays of Γ, hence if Γ is a perfect formulation for some integer set, vertices of Y will have integer γ. Further, even if Γ is not a perfect formulation, this shows that to find solutions with integer γ one need only consider cuts on Γ and not the entire polyhedron Y. For ease of notation, for $y \in Y$ define y_Γ to be the components of y in Γ. Finally, we note that a version of Theorem 7 appears as Lemma 5 in Faenza et al. (2010), 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_Γ is a vertex of Γ and for each extreme ray $r \in R$, r_Γ is an extreme ray of Γ. That is, $\text{proj}_\gamma(Y) = \Gamma$.*

Proof. Let $y \in Y$ such that $y = (x, x_1, \ldots, x_m, \gamma_1, \ldots, \gamma_m)$ and define $\gamma := y_\Gamma$. Since Γ 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

$$\gamma_i = \sum_{k=1}^p \lambda_k v_i^k + \sum_{l=1}^q \mu_l r_i^l, \text{ with } \sum_{k=1}^p \lambda_k = 1, \ \forall i \in [m] \tag{36}$$

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

$$x_i^k := \begin{cases} x_i v_i^k/\gamma_i \text{ if } i \in I_+ \\ x_i \text{ if } i \in I_0 \text{ and } \lambda_k > 0 \\ \hat{x}_i^k \in v_i^k P^i \text{ if } i \in I_0 \text{ and } \lambda_k = 0 \end{cases} \qquad \forall k \in [p], \tag{37}$$

$$x_i^l := \begin{cases} x_i r_i^l/\gamma_i \text{ if } i \in I_+ \\ 0 \text{ if } i \in I_0 \text{ and } \mu_l > 0 \\ \hat{x}_i^l \in r_i^l P^i \text{ if } i \in I_0 \text{ and } \mu_l = 0 \end{cases} \qquad \forall l \in [q], \tag{38}$$

and $x^k = \sum_{i=1}^m x_i^k$ for $k \in [p]$ and $x^l = \sum_{i=1}^m x_i^l$ for $l \in [q]$. For $k \in [p]$ define $y^k := (x^k, x_1^k, \ldots, x_m^k, v_1^k, \ldots, v_m^k)$ and for $l \in [q]$ define $y^l := (x^l, x_1^l, \ldots, x_m^l, r_1^l, \ldots, r_m^l)$.

We first check the feasibility of the points constructed above. So for each $k \in [p]$, consider y^k. By construction $y_\Gamma^k \in \Gamma$ and $x^k = \sum_{i=1}^m x_i^k$, so for feasibility we need verify that $A^i x_i^k \leq v_i^k b^i$. Suppose $i \in I_+$, then $A^i x_i \leq \gamma_i b^i$, and multiplying both sides by v_i^k and dividing by γ_i shows x_i^k is feasible. Now suppose $i \in I_0$ and so $A^i x_i \leq 0$. If $\lambda_k > 0$, then we must have $v_i^k = 0$, so x_i^k is feasible. If $\lambda_k = 0$, x_i^k 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 $\sum_{k=1}^p \lambda_k x_i^k + \sum_{l=1}^q \mu_l x_i^l = \sum_{k=1}^p \lambda_k x_i v_i^k/\gamma_i + \sum_{l=1}^q \mu_l x_i r_i^l/\gamma_i = \frac{x_i}{\gamma_i}(\sum_{k=1}^p \lambda_k v_i^k + \sum_{l=1}^q \mu_l r_i^l) = x_i$. Conversely, suppose $i \in I_0$, then $\sum_{k=1}^p \lambda_k x_i^k + \sum_{l=1}^q \mu_l x_i^l = \sum_{k:\lambda_k>0} \lambda_k x_i + \sum_{k:\lambda_k=0} \lambda_k \hat{x}_i^k + \sum_{l:\mu_l>0} \mu_l 0 + \sum_{l:\mu_l=0} \mu_l \hat{x}_i^l = \sum_{k:\lambda_k>0} \lambda_k x_i + 0 + 0 + 0 = x_i \sum_{k:\lambda_k>0} \lambda_k = x_i$. It then follows, $\sum_{k=1}^p \lambda_k x^k + \sum_{l=1}^q \mu_l x^l = \sum_{k=1}^p \lambda_k \sum_{i=1}^m x_i^k + \sum_{l=1}^q \mu_l \sum_{i=1}^m x_i^l = \sum_{i=1}^m (\sum_{k=1}^p \lambda_k x_i^k + \sum_{l=1}^q \mu_l x_i^l) = \sum_{i=1}^m x_i = x$. Hence, we have shown $y = \sum_{k=1}^p \lambda_k y^k + \sum_{l=1}^q \mu_l y^l$ with $\lambda, \mu \geq 0$ and $\sum_{k=1}^p \lambda_k = 1$, proving the theorem. □

As mentioned, Theorem 7 demonstrates that if Γ 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 $S\big|_{\mathbb{Z}_+} = \bigcup_{\gamma \in \Gamma \cap \mathbb{Z}_+} \{\sum_{i=1}^m \gamma_i P^i\}$. Noting that the polyhedron of Theorem 1 is exactly of this form, we see that the vertices of the polytope D must have integer γ.