SAND2018-11765J

A Novel Matching Formulation for Startup Costs in Unit Commitment*

Bernard Knueven[†1], James Ostrowski[‡2] and Jean-Paul Watson[§3]

[1]Discrete Math & Optimization, Sandia National Laboratories, Albuquerque, NM 87185
[2]Department of Industrial and Systems Engineering, University of Tennessee, Knoxville, TN 37996
[3]Data Science & Cyber Analytics, Sandia National Laboratories, Albuquerque, NM 87185

September 18, 2018

Abstract

We present a novel formulation for startup cost computation in the unit commitment problem (UC). Both our proposed formulation and existing formulations in the literature are placed in a formal, theoretical dominance hierarchy based on their respective linear programming relaxations. Our proposed formulation is tested empirically against existing formulations on large-scale UC instances drawn from real-world data. While requiring more variables than the current state-of-the-art formulation, our proposed formulation requires fewer constraints, and is empirically demonstrated to

*B. Knueven and J.-P. Watson were supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract number KJ0401000 through the project "Multifaceted Mathematics for Complex Energy Systems", and the Grid Modernization Initiative of the U.S. DOE, under project 1.4.26, as part of the Grid Modernization Laboratory Consortium, a strategic partnership between DOE and the national laboratories to bring together leading experts, technologies, and resources to collaborate on the goal of modernizing the nation's grid. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energys National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. J. Ostrowski was supported by NSF award number 1332662 and by DOE (ASCR) award DE-SC0018175.

†bknueve@sandia.gov
‡jostrows@utk.edu
§jwatson@sandia.gov

be as tight as a perfect formulation for startup costs. This tightening can reduce the computational burden in comparison to existing formulations, especially for UC instances with large reserve margins and high penetration levels of renewables.

1 Nomenclature

1.1 Indices and Sets

$g \in \mathcal{G}$	Thermal generators
$l \in \mathcal{L}_g$	Piecewise production cost intervals for generator g: $1, \ldots, L_g$.
$s \in \mathcal{S}_g$	Startup categories for generator g, from hottest (1) to coldest (S_g).
$t \in \mathcal{T}$	Hourly time steps: $1, \ldots, T$.

1.2 Parameters

c_g^l	Cost coefficient for piecewise segment l for generator g (\$/MWh).
c_g^s	Startup cost in category s for generator g (\$).
c_g^u	Cost of generator g running and operating at minimum production $\underline{P}_g$ (\$/h).
$D(t)$	Load (demand) at time t (MW).
DT_g	Minimum down time for generator g (h).
$\overline{P}_g$	Maximum power output for generator g (MW).
$\overline{P}_g^l$	Maximum power for piecewise segment l for generator g (MW).
$\underline{P}_g$	Minimum power output for generator g (MW).
$R(t)$	Spinning reserve at time t (MW).
RD_g	Ramp-down rate for generator g (MW/h).
RU_g	Ramp-up rate for generator g (MW/h).
SD_g	Shutdown rate for generator g (MW/h).
SU_g	Startup rate for generator g (MW/h).
TC_g	Time down after which generator g goes cold, i.e., enters state S_g.
$\underline{T}_g^s$	Time offline after which the startup category s is available ($\underline{T}_g^1 = DT_g$, $\underline{T}_g^{S_g} = TC_g$).
$\overline{T}_g^s$	Time offline after which the startup category s is no longer available ($= \underline{T}_g^{s+1}$, $\overline{T}_g^{S_g} = +\infty$).
UT_g	Minimum up time for generator g (h).
$W(t)$	Aggregate renewable generation available at time t (MW).

1.3 Variables

$p_g(t)$	Power above minimum for generator g at time t (MW), ≥ 0.
$p_W(t)$	Aggregate renewable generation used at time t (MW), ≥ 0.
$p_g^l(t)$	Power from piecewise interval l for generator g at time t (MW), ≥ 0.
$r_g(t)$	Spinning reserves provided by generator g at time t (MW), ≥ 0.

$u_g(t)$	Commitment status of generator g at time t, $\in \{0,1\}$.
$v_g(t)$	Startup status of generator g at time t, $\in \{0,1\}$.
$w_g(t)$	Shutdown status of generator g at time t, $\in \{0,1\}$.
$c_g^{SU}(t)$	Startup cost for generator g at time t (\$), ≥ 0.
$\delta_g^s(t)$	Startup in category s for generator g at time t, $\in \{0,1\}$.
$x_g(t,t')$	Indicator arc for shutdown at time t, startup at time t', uncommitted for $i \in [t,t')$, for generator g, $\in \{0,1\}$.
$y_g(t,t')$	Indicator arc for startup at time t, shutdown at time t', committed for $i \in [t,t')$, for generator g, $\in \{0,1\}$.

2 Introduction

The unit commitment problem (UC) concerns the scheduling of thermal generators to meet projected demand while minimizing system operations cost [34]. Here, we propose a new formulation for representing thermal generator startup costs, which leads to a tightening of the linear programming (LP) relaxation of the mixed-integer linear programming (MILP) UC problem. We then empirically demonstrate that the tighter LP relaxation can translate into reduced run-times to solve the MILP UC using commercial branch-and-cut solvers.

MILP formulations for UC have been of interest since Garver's original formulation [8]. These are extremely difficult problems to solve in practice, e.g., at the scale of the Midcontinent Independent System Operator in the United States. Many practical problems involve hundreds to thousands of generators and a time horizon of at least 48 hours. Further, solutions must be computed in tens of minutes at most. As a consequence, system operators often have to use substantially suboptimal solutions to comply with the time limit, i.e., with optimality gaps that are sometimes tens of percents [3].

There are a few approaches for reducing run-times to an optimal UC solution. One approach is via decomposition. The intuition is that loosely-connected parts of the UC problem can be decomposed into easier subproblems, and a solution to the original can be developed through an iterative process. One way to decompose UC is by generators – splitting the generator set $\mathcal{G}$ into subsets (by location or other criteria). Classical decomposition methods (such as ADMM) can then be used to force convergence between subproblems [6, 28]. Another possible decomposition is on the time horizon – the principle here being that after a sufficiently long period decisions made previously do not have much affect on decisions made now. Such an approach is explored in [12].

An alternative approach for reducing run times is stronger formulations for UC, and this research has found its way into practice. Most of this work has focused on tightening the polyhedral description of a single generator's dispatch. In [16] an exponential convex hull description for minimum up and down times in terms of a generator's status variables is given; [27] uses the startup and shutdown status variables to describe the same set using only a linear number of inequalities. This result is extended in [10] to generators with startup and shutdown power constraints. Inequalities to tighten the formulation of the ramping process are considered in [5, 13, 21, 23].

A formulation for time-dependent startup costs based on generator commitment variables appears in [20]; [2] considers the same formulation in the context of a MILP approach to UC. Startup cost categories together with associated indicator variables are introduced in [19]. [17] improves on the indicator formulation from [30] and demonstrates empirically that the use of startup category indicators results in a tighter formulation than those described in [20] and [2]. [18] uses this same approach to model generator start-up and shut-down energy production. [1] shows that the epigraph for concave non-decreasing startup costs modeled using generator status variables has an exponential number of facets. However, [1] provides a linear-time separation algorithm for computing these facets. Finally, a restrictive temperature-based model for startup cost is presented in [29].

In this paper we introduce a novel matching formulation for time-dependent startup costs in UC. We theoretically analyze the strength of our formulation relative to existing formulations in the literature, and introduce an additional formulation as an intermediary to ease the comparison between existing formulations. We then empirically analyze the impact of our new formulation, both in an absolute sense and relative to other formulations, on the ability of commercial branch-and-cut software packages to solve utility-scale UC problems.

The remainder of this paper is organized as follows. We begin in Section 3 with a discussion of the base UC formulation, without startup cost components. Section 4 then details both existing and two novel startup cost formulations for UC. In Section 5, we establish a provable dominance hierarchy concerning the relative tightness of LP relaxations for the different startup cost formulations. We empirically compare the performance of the various startup cost formulations in Section 6, using large-scale UC instances based on industrial data. We discuss the implications of our results in Section 7. Finally, we conclude with a summary of our contributions in Section 8.

3 Unit Commitment Formulation

We present a MILP UC formulation based on [17] that we will use as the baseline for our comparison between startup cost formulations. We assume that the production cost is piecewise linear convex in $p_g(t)$, where L_g is the number of piecewise intervals and $\overline{P}_g^0 = \underline{P}_g$ is the start of the first interval. Let $\mathcal{G}^1$ be the subset of generators that have $UT^g = 1$ and $\mathcal{G}^{>1}$ be the subset of generators with $UT^g > 1$. We then formulate the UC problem as follows:

$$\min \sum_{g\in\mathcal{G}}\sum_{t\in\mathcal{T}} \left(\sum_{l\in\mathcal{L}_g} (c_g^l p_g^l(t)) + c_g^u u_g(t) + c_g^{SU}(t) \right) \tag{1a}$$

subject to:

$$\sum_{g\in\mathcal{G}} \big(p_g(t) + \underline{P}_g u_g(t)\big) + p_W(t) = D(t) \qquad \forall t\in\mathcal{T} \tag{1b}$$

$$\sum_{g\in\mathcal{G}} r_g(t) \geq R(t) \qquad \forall t\in\mathcal{T} \tag{1c}$$

$$\begin{aligned} p_g(t)+r_g(t) \leq{} & (\overline{P}_g-\underline{P}_g)u_g(t) \\ & -(\overline{P}_g-SU_g)v_g(t) \end{aligned} \qquad \forall t\in\mathcal{T},\forall g\in\mathcal{G}^1 \tag{1d}$$

$$\begin{aligned} p_g(t)+r_g(t) \leq{} & (\overline{P}_g-\underline{P}_g)u_g(t) \\ & -(\overline{P}_g-SD_g)w_g(t+1) \end{aligned} \qquad \forall t\in\mathcal{T},\forall g\in\mathcal{G}^1 \tag{1e}$$

$$\begin{aligned} p_g(t)+r_g(t) \leq{} & (\overline{P}_g-\underline{P}_g)u_g(t) \\ & -(\overline{P}_g-SU_g)v_g(t) \\ & -(\overline{P}_g-SD_g)w_g(t+1) \end{aligned} \qquad \forall t\in\mathcal{T},\forall g\in\mathcal{G}^{>1} \tag{1f}$$

$$p_g(t)+r_g(t)-p_g(t-1) \leq RU_g \qquad \forall t\in\mathcal{T},\forall g\in\mathcal{G} \tag{1g}$$

$$p_g(t-1)-p_g(t) \leq RD_g \qquad \forall t\in\mathcal{T},\forall g\in\mathcal{G} \tag{1h}$$

$$p_g(t) = \sum_{l\in\mathcal{L}_g} p_g^l(t) \qquad \forall t\in\mathcal{T},\forall g\in\mathcal{G} \tag{1i}$$

$$p_g^l(t) \leq (\overline{P}_g^l-\overline{P}_g^{l-1})u_g(t) \qquad \forall t\in\mathcal{T},\forall l\in\mathcal{L}_g,\ \forall g\in\mathcal{G} \tag{1j}$$

$$u_g(t)-u_g(t-1) = v_g(t)-w_g(t) \qquad \forall t\in\mathcal{T},\forall g\in\mathcal{G} \tag{1k}$$

$$\sum_{i=t-UT_g+1}^{t} v_g(i) \leq u_g(t) \qquad \forall t\in[UT_g,T],\forall g\in\mathcal{G} \tag{1l}$$

$$\sum_{i=t-DT_g+1}^{t} w_g(i) \leq 1-u_g(t) \qquad \forall t\in[DT_g,T],\forall g\in\mathcal{G} \tag{1m}$$

$$p_W(t) \leq W(t) \qquad \forall t\in\mathcal{T} \tag{1n}$$

$$p_g^l(t) \in \mathbb{R}_+ \qquad \forall t\in\mathcal{T},\forall l\in\mathcal{L}_g,\forall g\in\mathcal{G} \tag{1o}$$

$$p_g(t),r_g(t) \in \mathbb{R}_+ \qquad \forall t\in\mathcal{T},\forall g\in\mathcal{G} \tag{1p}$$

$$p_W(t) \in \mathbb{R}_+ \qquad \forall t\in\mathcal{T} \tag{1q}$$

$$u_g(t),v_g(t),w_g(t) \in \{0,1\} \qquad \forall t\in\mathcal{T},\forall g\in\mathcal{G}. \tag{1r}$$

Constraints (1b – 1r) are standard in UC formulations without time-varying startup costs [13, 17].

We will take the formulation above as given, and for the remainder of the paper we will focus on the formulation of the startup cost $c_g^{SU}(t)$.

4 Startup Cost Formulations

In this section, we introduce the formulations for startup cost $c_g^{SU}(t)$ examined in this paper. For notational ease, since in all cases we are referencing a single generator, we will drop the subscript g on all variables and parameters in this section and in the following section.

4.1 Formulations from the Literature

4.1.1 One Binary Formulation (1-bin)

The typical formulation for startup costs using only the status variable u is [2,20]

$$c^{SU}(t) \geq c^s \left(u(t) - \sum_{i=1}^{\underline{T}^s} u(t-i) \right) \qquad \forall s \in \mathcal{S},\ \forall t \in \mathcal{T} \tag{2a}$$

$$c^{SU}(t) \geq 0 \qquad \forall t \in \mathcal{T}. \tag{2b}$$

This formulation has the advantage of only needing as many constraints as startup types, and no additional variables.

4.1.2 Strengthened One Binary Formulation (1-bin*)

As pointed out in [29], the 1-bin formulation above can be strengthened by increasing the coefficients on the $u(t-i)$ variables

$$\begin{aligned} c^{SU}(t) \geq c^s & \left(u(t) - \sum_{i=1}^{DT} u(t-i) \right) \\ & - \sum_{k=1}^{s-1} \left((c^s - c^k) \sum_{i=\underline{T}^k+1}^{\overline{T}^k} u(t-i) \right) \qquad \forall s \in \mathcal{S},\ \forall t \in \mathcal{T} \end{aligned} \tag{3a}$$

$$c^{SU}(t) \geq 0 \qquad \forall t \in \mathcal{T}. \tag{3b}$$

4.1.3 Startup Type Indicator Formulation (STI)

The formulation proposed in [17] introduces binary indicator variables for each startup type. Specifically, for each startup type s, we have $\delta^s(t)$, $\forall t \in \mathcal{T}$, which is 1 if the generator has a type s startup in time t and 0 otherwise. The corresponding constraints are

$$\delta^s(t) \leq \sum_{i=\underline{T}^s}^{\overline{T}^s-1} w(t-i) \qquad \forall s \in \mathcal{S} \setminus \{S\},\ \forall t \in \mathcal{T} \tag{4a}$$

$$v(t) = \sum_{s=1}^{S} \delta^s(t) \qquad \forall t \in \mathcal{T}. \tag{4b}$$

We can replace the objective function variables $c^{SU}(t)$ using the substitution

$$c^{SU}(t) = \sum_{s=1}^{S} c^s \delta^s(t) \qquad \forall t \in \mathcal{T}. \tag{4c}$$

4.1.4 Extended Formulation (EF)

The authors of [26] propose an extended formulation for startup and shutdown sequences, which provides a perfect formulation for startup costs, in the space of binary variables. We call a formulation *perfect* if the vertices of the polytope described by the formulation are integer. Note that if other variables or constraints are added, a formulation may lose this property. Let $y(t,t')=1$ if there is a startup in time t and a shutdown in time t' and 0 otherwise, for $t' \geq t+UT$. Similarly let $x(t,t')=1$ if there is a shutdown in time t and a startup in time t' and 0 otherwise, for $t' \geq t+DT$. The constraints are

$$\sum_{\{t'|t'>t\}} y(t,t')=v(t) \qquad \forall t\in\mathcal{T} \tag{5a}$$

$$\sum_{\{t'|t'<t\}} y(t',t)=w(t) \qquad \forall t\in\mathcal{T} \tag{5b}$$

$$\sum_{\{t'|t'<t\}} x(t',t)=v(t) \qquad \forall t\in\mathcal{T} \tag{5c}$$

$$\sum_{\{t'|t'>t\}} x(t,t')=w(t) \qquad \forall t\in\mathcal{T} \tag{5d}$$

$$\sum_{\{\tau,\tau'|\tau\leq t<\tau'\}} y(\tau,\tau')=u(t) \qquad \forall t\in\mathcal{T}. \tag{5e}$$

Note that with constraints (5a–5e), constraints (1k–1m) become redundant. Hence, the u, v, and w variables may be eliminated along with (1k–1m) while not losing validity or strength. The startup costs are calculated by placing the appropriate coefficient on the x variables

$$c^{SU}(t)=\sum_{s=1}^{S} c^s \left(\sum_{t'=t-\overline{T}^s+1}^{t-\underline{T}^s} x(t',t) \right) \qquad \forall t\in\mathcal{T}, \tag{5f}$$

where the inside summation is understood to be taken over valid t'.

From integer programming theory [33], we know this formulation to be integral because it is a network flow model, where the vertices are two partite sets, one for startups and the other for shutdowns, and the arcs y connect startups to feasible shutdowns and the arcs x connect shutdowns to feasible startups. By putting a flow of one unit through the network, we arrive at a feasible generator schedule. Note integrality comes at the cost of needing $\mathcal{O}(|\mathcal{T}|^2)$ additional variables to model startup costs.

4.2 Novel Formulations

Here we present two new formulations for startup costs. The first can be seen as a relaxation of EF, and the second as 1-bin* with the inequalities strengthened by using the startup/shutdown indicators v and w.

4.2.1 Matching Formulation (Match)

Similar to EF, for $t \in \mathcal{T}$ let $x(t',t)=1$ if there is a shutdown in time t' and a startup in time t and 0 otherwise, for $t' \in \mathcal{T}$ such that $t-TC<t' \leq t-DT$. Note that this only requires $(TC-DT)|\mathcal{T}|$ additional variables. The associated constraints are

$$\sum_{t'=t-TC+1}^{t-DT} x(t',t) \leq v(t) \qquad \forall t \in \mathcal{T} \tag{6a}$$

$$\sum_{t'=t+DT}^{t+TC-1} x(t,t') \leq w(t) \qquad \forall t \in \mathcal{T} \tag{6b}$$

(where again the sums are understood to be taken over valid t'), and the objective function is

$$c^{SU}(t)=c^S v(t)+\sum_{s=1}^{S-1}(c^s-c^S)\left(\sum_{t'=t-\overline{T}^s+1}^{t-\underline{T}^s} x(t',t)\right) \qquad \forall t \in \mathcal{T}. \tag{6c}$$

Note that if $v(t)$ and $w(t)$ are already determined, these equations serve to match shutdowns with startups. That is, if $v(t)=1$ and $w(t')=1$, then in any optimal solution $x(t',t)=1$ since $c^s-c^S<0$. We arrive at this formulation by eliminating the arcs y from EF and the arcs $x(t',t)$ such that $t-t' \geq TC$.

4.2.2 Three Binary Formulation (3-bin)

This formulation is similar in spirit to the 1-bin* formulation, only instead of using the status variables u, we use the startup/shutdown variables v and w to keep track of the different types of startups, as follows:

$$c^{SU}(t) \geq c^s v(t)-\sum_{k=1}^{s-1}\left((c^s-c^k)\sum_{i=\underline{T}^k}^{\overline{T}^k-1} w(t-i)\right) \qquad \forall s \in \mathcal{S},\ \forall t \in \mathcal{T} \tag{7a}$$

$$c^{SU}(t) \geq 0 \qquad \forall t \in \mathcal{T}. \tag{7b}$$

Equation (7a) works analogously to equation (3a). That is, if we did not shutdown in the last $\overline{T}^s$ time periods, (7a) ensures that we pay at least c^s for a startup in time t. Note that when $s=1$, the second term is an empty sum, and hence is 0. Equation (7b) ensures that the startup cost is never negative. Note that relative to STI, this formulation needs the same number of constraints and has fewer variables (only the additional $|\mathcal{T}|$ variables for $c^{SU}(t)$, which can be eliminated for STI, while not needing the indicator variables δ). This formulation is presented to ease the comparison between STI and the 1-bin formulations.

5 Dominance Hierarchy of Startup Cost Formulations

In this section we establish relationships between the six formulations presented in Section 4. First, we consider the relationship between the tightness of each formulation. Let z_{EF}, z_{Match}, z_{STI}, z_{3bin}, z_{1bin^*}, and z_{1bin} be the linear programming relaxation values for the respective formulations. (Recall we are always interested in a minimization problem.) A basic assumption we need is that startup costs are non-decreasing, that is, for every $g \in \mathcal{G}$, $c_g^s \leq c_g^{s+1}$, $\forall s \in \mathcal{S}_g \setminus \{S_g\}$. The formulations presented in Section 4 are invalid without some version of this assumption, except for EF. In practice it is a safe assumption because the heat required to restart a generator is an increasing function of time, and so the total cost to restart a generator will also be increasing in time. We have the following:

Theorem 1. *When startup costs are non-decreasing,*

$$z_{1bin} \leq z_{1bin^*} \leq z_{3bin} \leq z_{STI} \leq z_{Match} \leq z_{EF},$$

that is, EF is the tightest formulation, 1-bin is the weakest formulation, with the relationship above amongst the others.

Proof. Since the EF formulation is the convex hull description, it is clear that it is the tightest formulation, implying $z_{Match} \leq z_{EF}$. Furthermore, [29] shows that 1bin* is a tighter formulation than 1bin, implying that $z_{1bin} \leq z_{1bin^*}$. As a result, we only need to prove the inner three binary relationships. To prove these relationships, i.e., that $z_A \leq z_B$, it is sufficient to show: (1) there is a linear mapping from the polytope associated with B onto the polytope associated with A that preserves objective value, and (2) that through this linear mapping, every constraint in formulation A is implied by constraints in formulation B. This is sufficient to show that $z_A \leq z_B$ as it shows that *all* feasible solutions for B can be mapped to solutions feasible for A with the same objective value.

$\mathbf{z_{STI} \leq z_{Match}}$: We proceed by demonstrating that all the inequalities in STI are implied by the inequalities in Match. First, consider the linear transformation from Match to STI

$$\delta^s(t) = \sum_{t'=t-\overline{T}^s+1}^{t-\underline{T}^s} x(t',t) \qquad \forall s \in \mathcal{S} \setminus \{S\}, \forall t \in \mathcal{T} \tag{8}$$

$$\delta^S(t) = v(t) - \sum_{t'=t-TC+1}^{t-DT} x(t',t) \qquad \forall t \in \mathcal{T}. \tag{9}$$

The equality constraints (4b) follow directly from the sum of (8) and (9). To see (4a), notice that by (6b),

$$x(i,t) \leq w(i) \qquad \forall t \in \{i+DT,...,i+TC-1\}, \forall i \in \mathcal{T}. \tag{10}$$

By (8) and (10), we have

$$\delta^s(t) \le \sum_{i=t-\overline{T}^s+1}^{t-\underline{T}^s} w(i) = \sum_{i=\underline{T}^s}^{\overline{T}^s-1} w(t-i) \qquad \forall s \in \mathcal{S} \setminus S,\ \forall t \in \mathcal{T}, \tag{11}$$

which is just (4a).

$\mathbf{z_{3bin}} \le \mathbf{z_{STI}}$: This follows by eliminating the indicators δ from the objective function using (4a) and (4b). As $c^k \ge c^s$ for all $k \in \mathcal{S}$ such that $k > s$, we have

$$\begin{aligned} c^{SU}(t) &= \sum_{k=1}^{S} c^k \delta^k(t) \\ &\ge \sum_{k=1}^{s-1} c^k \delta^k(t) + c^s \sum_{k=s}^{S} \delta^k(t) \\ &= c^s \sum_{k=1}^{S} \delta^k(t) - \sum_{k=1}^{s-1} (c^s - c^k) \delta^k(t) \\ &\ge c^s v(t) - \sum_{k=1}^{s-1} \left((c^s - c^k) \sum_{i=\underline{T}^k}^{\overline{T}^k-1} w(t-i) \right) \qquad \forall s \in \mathcal{S},\ \forall t \in \mathcal{T}, \end{aligned} \tag{12}$$

which is (7a). (7b) follows from the non-negativity of c^s and $\delta^s(t)$.

$\mathbf{z_{1bin^*}} \le \mathbf{z_{3bin}}$: (3b) and (7b) are the same, so we need show that (7a) implies (3a). Consider the inequality (7a), noting $v(t) \ge u(t) - u(t-1)$ by (1k) and $-w(t) \ge -u(t-1)$ by (1k) and (1l)

$$\begin{aligned} c^{SU}(t) &\ge c^s v(t) - \sum_{k=1}^{s-1} \left((c^s - c^k) \sum_{i=\underline{T}^k}^{\overline{T}^k-1} w(t-i) \right) \\ &\ge c^s (u(t) - u(t-1)) - \sum_{k=1}^{s-1} \left((c^s - c^k) \sum_{i=\underline{T}^k}^{\overline{T}^k-1} u(t-1-i) \right) \\ &\ge c^s (u(t) - u(t-1)) - \sum_{k=1}^{s-1} \left((c^s - c^k) \sum_{i=\underline{T}^k+1}^{\overline{T}^k} u(t-i) \right) \\ &\ge c^s \left(u(t) - \sum_{i=1}^{DT} u(t-i) \right) - \sum_{k=1}^{s-1} \left((c^s - c^k) \sum_{i=\underline{T}^k+1}^{\overline{T}^k} u(t-i) \right) \qquad \forall s \in \mathcal{S},\ \forall t \in \mathcal{T}, \end{aligned} \tag{13}$$

which is (3a). □ □

We note that Theorem 1 does not establish strict dominance. We only guarantee that, for example, EF is no worse than Match in its linear programming relaxation. Some of these relationships hold with equality under non-decreasing startup costs.

Table 1: Size of the Formulations

Formulation	# variables	# constraints
1-bin	$\mathcal{O}(\lvert\mathcal{T}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert\lvert\mathcal{T}\rvert)$
1-bin*	$\mathcal{O}(\lvert\mathcal{T}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert\lvert\mathcal{T}\rvert)$
3-bin	$\mathcal{O}(\lvert\mathcal{T}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert\lvert\mathcal{T}\rvert)$
STI	$\mathcal{O}(\lvert\mathcal{S}\rvert\lvert\mathcal{T}\rvert)$	$\mathcal{O}(\lvert\mathcal{S}\rvert\lvert\mathcal{T}\rvert)$
Match	$\mathcal{O}((TC-DT)\lvert\mathcal{T}\rvert)$	$\mathcal{O}(\lvert\mathcal{T}\rvert)$
EF	$\mathcal{O}(\lvert\mathcal{T}\rvert^2)$	$\mathcal{O}(\lvert\mathcal{T}\rvert)$

Theorem 2. *Suppose startup costs are non-decreasing in time. Then* $z_{3bin}=z_{STI}$ *and* $z_{Match}=z_{EF}$.

We relegate the proof of Theorem 2 to the online supplement [14]. We additionally show that in the space of binary variables, the Match formulation is integer optimal when startup costs are non-decreasing. This makes Match interesting from both a theoretical and practical perspective – under certain restrictions on the objective function it returns an integer optimal solution, just like the EF. But because Match exploits the objective structure, it is able to use many fewer variables and constraints, which allows it to be practically useful. In particular, Match only grows linearly in time (as compared to quadratic for the EF), so for long time horizons it is much more computationally tractable, while still preserving guarantees on integrality. As we will see in Section 6, Match also preserves this edge over EF when embedded in a large UC problem.

Table 1 compares formulation size as a function of problem parameters for each startup formulation. Note we only consider the variables needed in addition to the baseline formulation (1).

6 Computational Experiments

The dominance hierarchy for the various startup cost formulations introduced in Section 5 establishes their relative tightness. We quantify tightness as the optimal objective function value for the LP relaxation of the UC problem with a given startup cost formulation. In the context of a MILP, tighter LP relaxations can lead to more efficient branch-and-cut search, due to increased fathoming opportunities. However, the size and structure of the underlying LP varies across startup cost formulations, and reductions in branch-and-cut search time (measured in terms of number of tree nodes explored) may be offset by the cost of solving the LP relaxations at each node. Further, formulation details interact with heuristics and other features of MILP solvers, often in unpredictable ways.

In this context, we now experimentally compare the performance of the range of startup cost formulations for UC, using two state-of-the-art commercial MILP solvers. We consider two sets of problem instances. The first set of instances are realistic instances derived from publicly available market and regulatory data obtained from the California Independent System Operator (CAISO) in the US. The second set is the FERC generator set [15] (which itself is based on data from the PJM Interconnection in the US), with demand, reserve, and wind scenarios based on publicly available data obtained from PJM for 2015 [24, 25].

The "CAISO" instances have 610 thermal generators, of which 410 are schedulable, i.e., not forced to run. Generators with quadratic cost curves were approximated using $L_g=2$. Five 48-hour demand scenarios were examined; demands were taken directly from CAISO historical data. Four of the demand scenarios are based on historical information, while "Scenario400" is a hypothetical scenario where wind supply is on average 40% of demand; the wind profile is constructed based on actual CAISO wind data, scaled appropriately. For each instance the reserve level was varied from 0%, 1%, 3%, and 5% of demand, resulting in a total of 20 test instances. We allow for the possibility of curtailment of wind generation by (1b) and (1n). Each generator has only two startup categories, i.e., $S_g=2$.

The "FERC" instances are based on two generator sets publicly available from the US Federal Energy Regulatory Commission (FERC): a "Summer" set of generators and a "Winter" set of generators [15]. We use the Summer set of generators for dates in April - September and the Winter set for the remaining dates. After (i) excluding generators with missing or negative cost curves, (ii) letting $UT_g=DT_g=1$ for generators g with missing up/down time data, and (iii) eliminating generators marked as wind (we consider wind power separately), the Summer and Winter sets respectively contain 978 and 934 generators. No data on startup or shutdown power limits was provided by FERC, so we assume $SU_g=SD_g=\underline{P}^g$. Similarly, FERC provided no data for cool-down times, so we set $TC_g=2DT_g$. All generators had at most two startup types, i.e., $S_g\leq 2$, and the piecewise production cost curves are based on market bids, such that $1\leq L_g\leq 10$.

For the FERC instances, we consider twelve 48-hour demand, reserve, and wind scenarios from 2015, one for each month. In 2015, wind generation accounted for 2% of the electricity supplied in PJM, so we created twelve additional "high-wind" scenarios by multiplying the wind data for 2015 by a constant factor of 15 to increase mean wind energy supply for the year to 30% of load. A recent study conducted for PJM suggests that in less than a decade, renewables could achieve 30% penetration rates in the interconnection [9]. Like the CAISO instances, we allow for the curtailment of wind generation.

The two test instance sets represent vastly different systems. The CAISO instances consist of mostly small, flexible generators. Of the 410 schedulable generators, only 20 have irredundant ramping constraints (i.e., $RU_g\geq(\overline{P}_g-\underline{P}_g)$ and $RD_g\geq(\overline{P}_g-\underline{P}_g)$). Therefore, for 390 of the generators (95% of the total), EF, together with the equations from (1), is a convex hull description of each generator's dispatch. These flexible generators account for 75% of schedulable capacity. For both the Summer and Winter FERC generator sets, such flexible generators only account for 50% of the fleet, and approximately 30% of schedulable capacity.

Table 2: Summary of computational experiments for CAISO instances using Gurobi. For time (s) and number of branch-and-cut (B&C) nodes we report the geometric mean across the 20 instances, including those which reach the wall-clock limit of 600 seconds.

Formulation	EF	Match	STI	3-bin	1-bin*	1-bin
Time (s)	370.5	**43.12**	52.84	91.43	600	600
# of times best	0	**11**	8	1	0	0
# of times 2nd	0	8	**11**	1	0	0
Max. time (s)	600	**130**	243	600	600	600
# of time outs	7	**0**	**0**	1	20	20
# B&C nodes	1.510	**13.50**	20.22	39.09	5914	5827

Table 3: Summary of computational experiments for CAISO instances using CPLEX. For time (s) and number of branch-and-cut (B&C) nodes we report the geometric mean across the 20 instances, including those which hit the wall-clock limit of 600 seconds.

Formulation	EF	Match	STI	3-bin	1-bin*	1-bin
Time (s)	261	48.0	**40.8**	62.9	600	600
# of times best	0	5	**11**	4	0	0
# of times 2nd	0	6	**8**	6	0	0
Max. time (s)	600	**110**	223	423	600	600
# of time outs	2	**0**	**0**	**0**	20	20
# of B&C nodes	3.60	**2.83**	4.75	32.6	15442	15210

Computational experiments were conducted on a Dell PowerEdge T620 with two Intel Xeon E5-2670 processors, for a total of 16 cores and 32 threads, and 256GB of RAM, running the Ubuntu 14.04.5 Linux operating system. The Gurobi 7.0.1 MILP solver was used for the experiments labeled "Gurobi", while the CPLEX 12.7.1.0 MILP solver was used for the experiment labeled "CPLEX". Both solvers were allowed to use all 32 threads in each experimental trial. Here we present summaries of the computational experiments; the full results are available in the online supplement [14].

6.1 CAISO Instances

We first consider the experimental results for the CAISO instances. For both Gurobi and CPLEX, we impose a wall-clock time limit of 600 seconds; all other settings were left at their defaults. In Tables 2 and 3, we summarize the results for these instances. For each UC formulation we report the geometric mean time to an optimal solution (Time (s)), the number of instances for which that method did best (# of times best), the number of instances for which that method did second best (# of times 2nd), the longest run time across the 20 instances (Max. time (s)), and the number of instances for which that method hit the 600 second time limit (# of time outs). When a solver times out for an instance, we substitute the time limit in the calculation for the geometric mean, leading to an underestimation when an instance fails to solve for a given UC formulation. In the last row, we report the shifted geometric mean number of branch-and-cut tree nodes explored by the solver, substituting the number of nodes explored when the solver hits the time limit. To compute the shifted geometric mean, we add 1 to each node count, so as to avoid multiplying by 0 when the solver identifies a solution at the root node. A bold-faced entry in a row denotes the startup cost variant that performed best for the given measure.

We immediately see that both of the 1-bin variants are not competitive, and in no case identify an optimal solution within the time limit, even after exploring a considerable number of branch-and-cut nodes. This is consistent with results reported recently in the UC literature. Gurobi identifies optimal solutions to the EF variant in approximately half the cases, and CPLEX identifies optimal solutions in all but two cases. However, for both solvers, the EF variant exhibits significantly larger run times – presumably due to the size of the LP formulation – than those observed for the Match, STI, or 3-bin variants.

Overall, 3-bin variant is not competitive with the Match and STI variants, and Gurobi times out for one instance. Using CPLEX, the 3-bin variant is often the best or second-best, but when it performs poorly 3-bin often takes much longer than the Match and STI variants.

Comparing the Match and STI variants, we can see that overall Gurobi performs better using the Match variant while CPLEX performs better using the STI variant. However, for both solvers, the Match variant has a significantly (approximately 50%) lower maximum time across the CAISO test instances, suggesting it may be a more robust UC formulation in practice. Additionally, for both the Match and STI variants, solution times generally grow with increases in reserve level; we refer to the detailed results in [14]. The latter observation has significant potential impact on stochastic unit commitment solvers, as we discuss further below in Section 7.

Turning to the number of branch-and-cut nodes explored, in the case of the 1-bin variants, the large number of nodes explored is consistent with the inability of the solver to identify optimal solutions within the specified time limit. Interestingly, Gurobi and CPLEX typically did not leave the root node processing phase within the 600 second time limit when considering the EF variant. Further, we note that the size of the EF formulation makes cut generation (and heuristics) at the root node more difficult. In the case of the Match and STI variants, both Gurobi and CPLEX identify an optimal solution at the root node for instances with relatively low reserve levels, with CPLEX finding a root node solution more often. However, as reserve levels increase, the number of nodes explored increases. Finally, we observe that the relatively few number of tree nodes explored with the tighter Match and STI variants indicates relatively few opportunities for parallelism, at least in terms of accelerating the tree search

Table 4: Computational results for CAISO instances: Relative Integrality Gap (%), geometric mean across 20 instances.

Formulation	EF	Match	STI	3-bin	1-bin*	1-bin
Gap (%)	0.008	0.008	0.033	0.033	1.525	1.569

process.

Finally, in Table 4 we report the geometric mean relative integrality gap for each startup cost formulation. For each instance, we compute the relative integrality gap by taking the best integer solution objective value found across all six formulations and both solvers, denoted z^*_{IP}, and the objective value of the LP relaxation for each instance (as computed by Gurobi after relaxing the binary variables), denoted z^*_{LP}; we then report $(z^*_{IP}-z^*_{LP})/z^*_{IP}$ as a percentage. First, we observe that the results in Table 4 are consistent with and empirically verify the correctness of Theorem 1. The 1-bin variants are significantly weaker than the other variants, with the relative integrality gap typically exceeding 1%. We also note that in all instances, the relative integrality gap (and hence LP relaxation) for the EF and Match variants is identical; an analogous situation is observed for the STI and 3-bin variants. Lastly, we note that the Match formulation typically closes 50-90% of the integrality gap relative to STI (74% in geometric mean), which explains its computational benefit despite the additional variables required.

6.2 FERC Instances

Because the FERC instances are larger and therefore likely more difficult than the CAISO instances, we increased the wall-clock time limit to 900 seconds. Further, for Gurobi, we set the `Method` parameter to 3 so Gurobi would use the non-deterministic concurrent optimizer to solve the root LP relaxations. The non-deterministic concurrent optimizer solves LPs by running primal and dual simplex on one thread each and a barrier plus crossover method on the remaining 14 threads, returning an optimal LP basis from whichever method returns first. All other settings for Gurobi were left at their defaults. CPLEX settings were preserved at their defaults. When describing the computational results below, we separate the instances into two categories: the results considering the 2% wind penetration levels observed in 2015 and hypothetical 30% wind penetration levels based on the same data.

In Tables 5 and 6 we summarize the computational experiments for both Gurobi and CPLEX for the FERC instances. Tables 5 and 6 report the same statistics for the FERC instances as Tables 2 and 3 did for the CAISO instances. First, we consider the 2% wind penetration instances, which are reported in part (a) of both tables. We observe that the 1-bin variants and EF are not competitive with the Match and STI variants. As was the case with the CAISO instances, Match performs best with Gurobi, whereas STI performs better with

Table 5: Summary of computational experiments for FERC instances using Gurobi. For time (s) and number of branch-and-cut (B&C) nodes we report the geometric mean across the 12 instances, including those which hit the wall-clock limit of 900 seconds.

(a) 2% Wind Penetration

Formulation	EF	Match	STI	3-bin	1-bin*	1-bin
Time (s)	702	**154**	218	267	712	739
# of times best	0	**6**	4	2	0	0
# of times 2nd	0	**6**	5	1	0	0
Max. time (s)	900	**411**	491	841	900	900
# of time outs	4	**0**	**0**	**0**	7	7
# of B&C nodes	1.00	**1.38**	5.91	9.03	67.5	50.8

(b) 30% Wind Penetration

Formulation	EF	Match	STI	3-bin	1-bin*	1-bin
Time (s)	808	**215**	391	401	799	804
# of times best	0	**8**	2	2	0	0
# of times 2nd	2	1	**6**	3	0	0
Max. time (s)	900	**648**	900	900	900	900
# of time outs	6	**0**	2	3	10	10
# of B&C nodes	1.00	**4.66**	51.7	78.2	142	130

CPLEX. CPLEX does not find these instances difficult, solving all 12 problems using the Match and STI variant at the root node. The 3-bin variant is occasionally the fastest method for CPLEX for a given instance, but mirroring the CAISO instances it has a significantly inferior worse-case solve time than either Match or STI.

Now consider the FERC instances with 30% wind penetration levels. Here, we see that only the Match variant able to solve all 12 instances within the time limit on both solvers. Examining the geometric mean solve time, we observe that the Match variant reduces the solve time relative to the STI variant by 45% for Gurobi, with a more moderate reduction for CPLEX. Overall, the 30% wind penetration level instances are noticeably more difficult than the 2% wind penetration instances. However, for Gurobi, the Match variant requires only 40% more computational time on average to solve the former, while the STI variant requires more than 80% additional computational time. The situation is similar for CPLEX, where the Match variant only needs 36% more computational time on average for 30% wind instances, whereas STI variant needs 85% more computational time.

Next, we consider the number of branch-and-cut tree nodes explored when solving each instance. Looking at Table 5, we observe that for the Match variant, Gurobi typically locates an optimal solution at the root node, or at least very

Table 6: Summary of computational experiments for FERC instances using CPLEX. For time (s) and number of branch-and-cut (B&C) nodes we report the geometric mean across the 12 instances, including those which hit the wall-clock limit of 900 seconds.

(a) 2% Wind Penetration

Formulation	EF	Match	STI	3-bin	1-bin*	1-bin
Time (s)	478	136	**114**	162	499	538
# of times best	0	2	**5**	**5**	0	0
# of times 2nd	0	4	**6**	2	0	0
Max. time (s)	900	222	**150**	737	900	900
# of time outs	1	**0**	**0**	**0**	4	5
# of B&C nodes	1.23	**1.00**	**1.00**	7.52	356	414

(b) 30% Wind Penetration

Formulation	EF	Match	STI	3-bin	1-bin*	1-bin
Time (s)	604	**185**	211	298	784	798
# of times best	0	**5**	2	**5**	0	0
# of times 2nd	1	3	**8**	0	0	0
Max. time (s)	900	**269**	900	900	900	900
# of time outs	2	**0**	1	3	10	10
# of B&C nodes	1.95	**1.94**	5.91	25.3	1171	1153

early in the tree search process. Overall, the number of tree nodes explored under the Match variant is significantly less than that under the STI variant; the latter in turn dominates, as expected, the 3-bin and 1-bin variants. Mirroring the results for CAISO instances, Gurobi does not exit root node processing on the EF variant – in all cases the root relaxation is solved, but the time limit is exhausted applying cuts and heuristics. For both 1-bin variants, Gurobi spends a significant amount of time during root node processing generating cuts, which is why for some instances a small number of nodes are explored before the time limit expires. Consistent with the increase in relative instance difficulty, Gurobi requires more nodes to identify an optimal solution in the case of 30% wind instances, but the increase is much less pronounced than for the STI or 3-bin variants.

Examining the node count summaries for CPLEX in Table 6, we observe that using the Match and STI variants the 2% wind instances are easy, never leaving the root node. Similar to the experience with Gurobi, for 30% wind instances there is only a modest increase in node count for the Match variant (only one instance does not solve at the root node), and larger but still modest increases for STI and 3-bin variants. When it solves, the EF variant does so at the root node, and the 1-bin variants need more enumeration, and usually hit the time limit while still exploring the tree.

Table 7: Computational results for FERC instances: Relative Integrality Gap (%), geometric mean across each of the 12 instances.

(a) 2% Wind Penetration

Formulation	EF	Match	STI	3-bin	1-bin*	1-bin
Gap (%)	0.068	0.068	0.075	0.075	0.911	0.911

(b) 30% Wind Penetration

Formulation	EF	Match	STI	3-bin	1-bin*	1-bin
Gap (%)	0.206	0.206	0.314	0.314	3.003	3.003

Finally, we report the relative integrality gap for each combination of instance and variant in Table 7, calculated in the same manner as those reported in Table 4. Mirroring the results for the CAISO instances, we observe empirical verification of Theorem 1. Further, the EF and Match variants have identical integrality gaps, as do the STI and 3-bin variants. Unlike as was observed for the CAISO instances, the 1-bin* variant is not significantly tighter than the 1-bin variant, and the Match variant typically only closes 0%-40% of the root gap over STI. This result is partially explained by the fact that approximately half of the generators are ramp-constrained – and even in the EF case, we are not using an ideal formulation for ramp-constrained generators. However, for the January and February 30% wind penetration level instances, the difference is significant (Match closes 95% of the root gap for January and 67% of the root gap for February over STI, see [14]). This is consistent with the result that only the Match and EF variants were able to solve these instances within the time limits on both solvers. For the 2% wind penetration instances the Match variant only closes 8% of the relative integrality gap on average versus STI. Interestingly, Table 5 shows the Match variant is still computationally competitive for Gurobi, but STI is able to outperform Match using CPLEX because the extra variables are not providing much in the way of additional tightness.

6.3 Statistical Analysis

A statistical analysis of the computational results was performed using the Wilcoxon signed-rank test [32] for both Gurobi and CPLEX. On Gurobi, across the entire test set (both CAISO and FERC), Match is superior to all the other formulations examined at the α=0.01 level. On the other hand, using CPLEX, though Match was faster than STI and 3-bin in mean solve time, these differences were not significant at the α=0.05 level. Further, on the "Low Wind" instances (those being the 2% Wind Penetration instances from FERC and the instances corresponding to historical dates from CAISO), the STI variant is able to outperform Match at the α=0.01 level, though the difference in magnitude is only 16.8 seconds. Finally, we note that for the "High Wind" instances (Scenario400 instances from CAISO and 30% Wind Penetration instances from

FERC) Match is 60.4 seconds faster in mean solve time, but this difference was not significant at the α=0.05 level. This is likely due to this test being underpowered at n=16. The full results of the statistical analysis are available in the online supplement [14].

7 Discussion

We now discuss the implications of the computational experiments described above. First, we note that both the CAISO and FERC test instances have $S_g \leq 2$ for all generators g, due to the data available. In real-world instances, a non-trivial number of generators may have $S_g > 2$. Our proposed Match formulation of startup costs in UC can model more startup categories – up to TC_g – by simply changing the objective coefficients. In contrast, with the exception of EF, all other startup cost formulations require additional variables and/or constraints.

As the experiments on the CAISO instances demonstrate, reserve requirements do have a significant impact on the difficulty of solving UC. For instances with a 0% or 1% reserve requirement, Gurobi is able to solve all instances using the Match formulation in under a minute. This is an interesting observation in the context of stochastic unit commitment [31], in which reserve levels for individual scenarios are minimal, as the scenarios themselves are intended to capture the range of uncertainties that may be encountered. Further, we note that effective decomposition techniques for solving stochastic UC problems – including progressive hedging [4] – repeatedly solve individual (and thus deterministic) scenario problems. Thus, we expect our Match formulation to significantly accelerate the solution of stochastic UCs.

Our experiments also demonstrate that with a modern MILP solver, commodity workstation hardware, and tight formulations, we can quickly solve utility-scale UC problems to very small (<0.01%) optimality gaps. In fact, the results for the CAISO instances suggest they could be solved to even tighter gaps than the Gurobi and CPLEX defaults, within the imposed time limit. Reduced optimality gaps are important to guarantee market fairness, i.e., to ensure that a cheaper generator is scheduled in place of a more expensive one. The ability to run to very small optimality gaps is also important in the context of scenario-based decomposition approaches to stochastic UC. Deterministic UC scenarios often have feasible solutions which are far away from optimal. Thus, imposition of a tighter optimality gap can significantly improve convergence of algorithms such as progressive hedging, by providing strong initial solutions of individual scenarios – which are used to guide subsequent iterations of the algorithm. Further, the ability of progressive hedging to generate high-quality lower bounds for stochastic UC is dependent on how tightly the scenario UC problems are solved [4, 7].

In the context of stochastic UC – where renewable energy supply is the main driver of uncertainty – it is interesting to note that for both systems the high-wind scenarios (“Scenario400” for CAISO and the 30% wind penetration instances for FERC) are significantly more difficult, independent of startup formulation. However, these instances are also where our proposed Match formulation shows the most improvement over STI. Across all 16 high-wind scenarios, when using Gurobi the Match variant exhibited a >44% improvement in geometric mean solve time, with a more modest >15% improvement using CPLEX, and a 57% improvement in geometric mean relative integrality gap. In comparison, on the other 28 instances, when using Gurobi the Match variant only showed a 20% improvement in geometric mean solve time, with a 20% degradation using CPLEX, and a 50% geometric mean relative integrality gap closure over STI. This is not surprising, given that more variability in net-load implies there will be more switches in generator status.

Finally, we comment on the "synthetic" UC instances from [2], which are extended via replication in [21] and again in [17]. These originate from a now dated genetic algorithm UC paper [11], which has no indication that these were drawn from real-world data. Compared to the generator sets gathered from CAISO and FERC, these instances have much less flexible capacity (less than 10% in all cases), which implies that the ramping process is a much bigger factor in adjusting to changes in demand than generator switching. Additionally, the replication of the same 8 or 10 generators induces artificial symmetry into the problem, which can confound the branch-and-cut process. Though modern commercial MILP solvers have sophisticated symmetry detection, they do not capture all the symmetry in UC [22]. These factors together imply that the synthetic instances are less likely to be impacted by improvements in startup cost formulations. Based on the instances in [21], we created twenty 48-hour UC instances, and tested the six startup cost formulations on the platform described in Section 6 using Gurobi. After 1800 seconds of wall-clock time, the Match, STI, and 3-bin variants were able to solve only 6 of the 20 instances, the EF variant was able to solve 2 of the 20 instances, and the 1-bin variants only 1 of the 20 instances. In geometric mean Match was only able to close 5% of the relative integrality gap over STI. The confounding symmetry and inflexibility in these instances makes it difficult to draw a distinction between the Match, STI, and 3-bin variants, though they all out-perform the 1-bin, 1-bin*, and EF variants.

8 Conclusions

We have presented a novel matching formulation for time-dependent startup costs in UC, and an additional compact formulation for time-dependent startup costs as an intermediary between the STI and the 1-bin formulations. We have formally placed these two new formulations, in addition to existing alternatives, in a formal dominance hierarchy based on the corresponding LP relaxations. We examined the computational efficacy of the various alternative formulations for time-dependent startup costs on large-scale unit commitment instances based on real-world data from the PJM and CAISO independent system operators in the US using two commercial MILP solvers. We find that the proposed matching formulation is computationally as effective on average than the current state-of-the-art formulation, and is computationally more effective for high-wind penetration scenarios. Additionally, we empirically demonstrated that the proposed matching formulation is as tight as the ideal formulation while being more compact.

References

[1] Brandenberg, R., Huber, M., Silbernagl, M.: The summed start-up costs in a unit commitment problem. EURO Journal on Computational Optimization pp. 1–36 (2015)

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

[3] Chen, Y.: Personal correspondence (2016). Principal Advisor at Midcontinent Independent System Operator

[4] Cheung, K., Gade, D., Silva-Monroy, C., Ryan, S.M., Watson, J.P., Wets, R.J.B., Woodruff, D.L.: Toward scalable stochastic unit commitment. Energy Systems **6**(3), 417–438 (2015)

[5] Damc-Kurt, P., Kkyavuz, S., Rajan, D., Atamtrk, A.: A polyhedral study of production ramping. Mathematical Programming pp. 1–31 (2015)

[6] Feizollahi, M.J., Costley, M., Ahmed, S., Grijalva, S.: Large-scale decentralized unit commitment. International Journal of Electrical Power & Energy Systems **73**, 97–106 (2015)

[7] Gade, D., Hackebeil, G., Ryan, S.M., Watson, J.P., Wets, R.J.B., Woodruff, D.L.: Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. Mathematical Programming **157**(1), 47–67 (2016)

[8] Garver, L.L.: 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 (1962)

[9] GE Energy: PJM renewable integration study. PJM Interconnection (2014)

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

[11] Kazarlis, S.A., Bakirtzis, A., Petridis, V.: A genetic algorithm solution to the unit commitment problem. IEEE Transactions on Power Systems **11**(1), 83–92 (1996)

[12] Kim, K., Botterud, A., Qiu, F.: Temporal decomposition for improved unit commitment in power system production cost modeling. ANL Technical Report (2017). ANL/MCS-P7073-0717

[13] Knueven, B., Ostrowski, J., Wang, J.: The ramping polytope and cut generation for the unit commitment problem. INFORMS Journal on Computing (2017). To appear

[14] Knueven, B., Ostrowski, J., Watson, J.P.: Online supplement for a novel matching formulation for startup costs in unit commitment (2018)

[15] Krall, E., Higgins, M., O'Neill, R.P.: RTO unit commitment test system. Federal Energy Regulatory Commission (2012)

[16] Lee, J., Leung, J., Margot, F.: Min-up/min-down polytopes. Discrete Optimization **1**(1), 77–85 (2004)

[17] Morales-España, G., Latorre, J.M., Ramos, A.: Tight and compact MILP formulation for the thermal unit commitment problem. IEEE Transactions on Power Systems **28**(4), 4897–4908 (2013)

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

[19] Muckstadt, J.A., Wilson, R.C.: An application of mixed-integer programming duality to scheduling thermal generating systems. IEEE Transactions on Power Apparatus and Systems (12) (1968)

[20] Nowak, M.P., Römisch, W.: Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty. Annals of Operations Research **100**(1-4), 251–272 (2000)

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

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

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

[24] PJM: PJM - ancillary services. http://pjm.com/markets-and-operations/ancillary-services.aspx (2016). Accessed: 2016-01-07

[25] PJM: PJM - system operations. http://www.pjm.com/markets-and-operations/ops-analysis.aspx (2016). Accessed: 2016-01-07

[26] Pochet, Y., Wolsey, L.A.: Production planning by mixed integer programming. Springer Science & Business Media (2006)

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

[28] Ramanan, P., Yildirim, M., Chow, E., Gebraeel, N.: Asynchronous decentralized framework for unit commitment in power systems. Procedia Computer Science **108**, 665–674 (2017)

[29] Silbernagl, M., Huber, M., Brandenberg, R.: Improving accuracy and efficiency of start-up cost formulations in MIP unit commitment by modeling power plant temperatures. IEEE Transactions on Power Systems **31**(4), 2578–2586 (2016)

[30] Simoglou, C.K., Biskas, P.N., Bakirtzis, A.G.: Optimal self-scheduling of a thermal producer in short-term electricity markets by MILP. IEEE Transactions on Power Systems **25**(4), 1965–1977 (2010)

[31] Takriti, S., Birge, J., Long, E.: A stochastic model for the unit commitment problem. IEEE Transactions on Power Systems **11**(3), 1497–1508 (1996)

[32] Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics bulletin **1**(6), 80–83 (1945)

[33] Wolsey, L.A.: Integer programming. Wiley (1998)

[34] Wood, A.J., Wollenberg, B.F., Sheblé, G.B.: Power Generation, Operation and Control. John Wiley & Sons (2013)

Online Supplement for

A Novel Matching Formulation for Startup Costs in Unit Commitment*

Bernard Knueven†1, James Ostrowski‡2 and Jean-Paul Watson§3

[1]Discrete Math & Optimization, Sandia National Laboratories, Albuquerque, NM 87185

[2]Department of Industrial and Systems Engineering, University of Tennessee, Knoxville, TN 37996

[3]Data Science & Cyber Analytics, Sandia National Laboratories, Albuquerque, NM 87185

September 18, 2018

Abstract

This document is an online supplement for [1]. Here we present a proof of Theorem 2, complete computational results used to make the summary tables in [1], and a statical analysis of the computational performance of the various formulations.

*B. Knueven and J.-P. Watson were supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract number KJ0401000 through the project "Multifaceted Mathematics for Complex Energy Systems", and the Grid Modernization Initiative of the U.S. DOE, under project 1.4.26, as part of the Grid Modernization Laboratory Consortium, a strategic partnership between DOE and the national laboratories to bring together leading experts, technologies, and resources to collaborate on the goal of modernizing the nation's grid. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energys National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. J. Ostrowski was supported by NSF award number 1332662 and by DOE (ASCR) award DE-SC0018175.

†bknueve@sandia.gov

‡jostrows@utk.edu

§jwatson@sandia.gov

1 Proof of Theorem 2

1.1 STI and 3-bin have the same LP relaxation values (under non-decreasing startup costs)

First, we'll prove $z_{3bin} = z_{STI}$. Take g and t as fixed for the proof, as the process can be repeated for each g and t. We will assume throughout that startup costs are non-decreasing in time. Consider

$$v(t) = \sum_{s=1}^{S} \delta^s(t) \tag{1a}$$

$$c^{SU}(t) \geq \sum_{s=1}^{S} c^s \delta^s(t) \tag{1b}$$

$$\delta^s(t) \leq \sum_{i=\underline{T}^s}^{\overline{T}^s-1} w(t-i) \qquad \forall s \in \mathcal{S} \setminus \{S\} \tag{1c}$$

$$\delta^s(t) \geq 0 \qquad \forall s \in \mathcal{S}, \tag{1d}$$

$$c^{SU}(t) \geq 0 \tag{1e}$$

which is the epigraph of the startup-cost at time t for the STI formulation. We will prove the theorem by projecting out the $\delta^s(t)$ variables using Fourier-Motzkin elimination and showing this gives exactly the 3-bin formulation. First, consider $\delta^S(t)$. This term is only in (1a), (1b), and (1d). Rearranging yields:

$$\delta^S(t) = v(t) - \sum_{s=1}^{S-1} \delta^s(t) \tag{2a}$$

$$c^S \delta^S(t) \leq c^{SU}(t) - \sum_{s=1}^{S-1} c^s \delta^s(t) \tag{2b}$$

$$\delta^S(t) \geq 0 \tag{2c}$$

Equations (2a) and (2c):

$$v(t) - \sum_{s=1}^{S-1} \delta^s(t) \geq 0, \tag{3a}$$

equations (2a) and (2b) yield:

$$c^{SU}(t) - \sum_{s=1}^{S-1} c^s \delta^s(t) \geq c^S \left(v(t) - \sum_{s=1}^{S-1} \delta^s(t) \right), \tag{3b}$$

and equations (2b) and (2c):

$$c^{SU}(t) - \sum_{s=1}^{S-1} c^s \delta^s(t) \geq 0. \tag{3c}$$

We can see that (3c) is redundant. This leaves the system:

$$v(t) \geq \sum_{s=1}^{S-1} \delta^s(t) \tag{4a}$$

$$c^{SU}(t) \geq c^S v(t) - \sum_{s=1}^{S-1} (c^S - c^s)\delta^s(t) \tag{4b}$$

$$\delta^s(t) \leq \sum_{i=\underline{T}^s}^{\overline{T}^s-1} w(t-i) \qquad \forall s \in \mathcal{S} \setminus \{S\} \tag{4c}$$

$$\delta^s(t) \geq 0 \qquad \forall s \in \{1, \dots, S-1\}, \tag{4d}$$

$$c^{SU}(t) \geq 0 \tag{4e}$$

Notice that if $S = 1$, we are done. So suppose $S > 1$. We can rearrange (4) by $\delta^{S-1}(t)$ (temporarily dropping those terms in which it does not appear):

$$\delta^{S-1}(t) \leq v(t) - \sum_{s=1}^{S-2} \delta^s(t) \tag{5a}$$

$$(c^S - c^{S-1})\delta^{S-1}(t) \geq c^S v(t) - \sum_{s=1}^{S-2} (c^S - c^s)\delta^s(t) - c^{SU}(t) \tag{5b}$$

$$\delta^{S-1}(t) \leq \sum_{i=\underline{T}^{S-1}}^{\overline{T}^{S-1}-1} w(t-i) \tag{5c}$$

$$\delta^{S-1}(t) \geq 0 \tag{5d}$$

Again we can eliminate. First notice that when startup costs are non-decreasing, $c^S - c^{S-1} \geq 0$. Equations (5a) and (5b) give:

$$c^{SU}(t) \geq c^{S-1} v(t) - \sum_{s=1}^{S-2} (c^{S-1} - c^s)\delta^s(t). \tag{6a}$$

Equations (5a) and (5d):

$$v(t) \geq \sum_{s=1}^{S-2} \delta^s(t), \tag{6b}$$

and equations (5b) and (5c) yields:

$$c^{SU}(t) \geq c^S v(t) - \sum_{s=1}^{S-2} (c^S - c^s)\delta^s(t) - (c^S - c^{S-1}) \sum_{i=\underline{T}^{S-1}}^{\overline{T}^{S-1}-1} w(t-i). \tag{6c}$$

Equations (5c) and (5d) just asserts the non-negativity of the w variables, thus eliminating $\delta^{S-1}(t)$. Again, if $S = 2$ we are done, otherwise, we can take this as a base case and proceed by induction. Consider the following Lemma.

Lemma 1. *Suppose after eliminating the last K startup indicators we have*

$$v(t) \geq \sum_{s=1}^{S-K} \delta^s(t) \tag{7a}$$

$$c^{SU}(t) \geq c^{S-k}v(t) - \sum_{s=1}^{S-K}(c^{S-k} - c^s)\delta^s(t) - \sum_{j=S-(K-1)}^{(S-k)-1}(c^{S-k} - c^j)\sum_{i=\underline{T}^j}^{\overline{T}^j-1} w(t-i) \quad \forall k \in \{0, \dots, K-1\} \tag{7b}$$

$$\delta^s(t) \leq \sum_{i=\underline{T}^s}^{\overline{T}^s-1} w(t-i) \quad \forall s \in \{1, \dots, S-K\} \tag{7c}$$

$$\delta^s(t) \geq 0 \quad \forall s \in \{1, \dots, S-K\} \tag{7d}$$

$$c^{SU}(t) \geq 0. \tag{7e}$$

Then after using Fourier-Motzkin to eliminate the $\delta^{S-K}(t)$ variable, we are left with the system

$$v(t) \geq \sum_{s=1}^{S-(K+1)} \delta^s(t) \tag{8a}$$

$$\begin{aligned} c^{SU}(t) \geq c^{S-k}v(t) &- \sum_{s=1}^{S-(K+1)}(c^{S-k} - c^s)\delta^s(t) \\ &- \sum_{j=S-((K+1)-1)}^{(S-k)-1}(c^{S-k} - c^j)\sum_{i=\underline{T}^j}^{\overline{T}^j-1} w(t-i) \quad \forall k \in \{0, \dots, K\} \end{aligned} \tag{8b}$$

$$\delta^s(t) \leq \sum_{i=\underline{T}^s}^{\overline{T}^s-1} w(t-i) \quad \forall s \in \{1, \dots, S-(K+1)\} \tag{8c}$$

$$\delta^s(t) \geq 0 \quad \forall s \in \{1, \dots, S-(K+1)\} \tag{8d}$$

$$c^{SU}(t) \geq 0. \tag{8e}$$

Proof. We will eliminate the $\delta^{S-K}(t)$ variable from (7). We can re-arrange and notice it appears in the following terms:

$$\delta^{S-K} \leq v(t) - \sum_{s=1}^{S-(K+1)} \delta^s(t) \tag{9a}$$

$$(c^{S-k} - c^{S-K})\delta^{S-K} \geq c^{S-k}v(t) - \sum_{s=1}^{S-(K+1)} (c^{S-k} - c^s)\delta^s(t)$$

$$- \sum_{j=S-(K-1)}^{(S-k)-1} (c^{S-k} - c^j) \sum_{i=\underline{T}^j}^{\overline{T}^j-1} w(t-i) - c^{SU}(t) \qquad \forall k \in \{0, \ldots, K-1\} \tag{9b}$$

$$\delta^{S-K} \leq \sum_{\underline{T}^{S-K}}^{\overline{T}^{S-K}-1} w(t-i) \tag{9c}$$

$$\delta^{S-K} \geq 0. \tag{9d}$$

We proceed with the Fourier-Motzkin elimination. Combining equations (9a) with (9b):

$$c^{SU}(t) \geq c^{S-K}v(t) - \sum_{s=1}^{S-(K+1)} (c^{S-K} - c^s)\delta^s(t) - \sum_{j=S-(K-1)}^{(S-k)-1} (c^{S-k} - c^j) \sum_{i=\underline{T}^j}^{\overline{T}^j-1} w(t-i) \quad \forall k \in \{0, \ldots, K-1\}. \tag{10a}$$

As the terms in the second sum of (10a) are non-negative (as startup costs are non-decreasing), we see that the inequality (10a) is strongest when this sum is empty, i.e., when $k = K - 1$, and all the others can be dropped, yielding:

$$c^{SU}(t) \geq c^{S-K}v(t) - \sum_{s=1}^{S-(K+1)} (c^{S-K} - c^s)\delta^s(t) \tag{10b}$$

Equations (9a) and (9d) give:

$$v(t) \geq \sum_{s=1}^{S-(K+1)} \delta^s(t). \tag{10c}$$

Equations (9b) with (9c) yields:

$$c^{SU}(t) \geq c^{S-k}v(t) - \sum_{s=1}^{S-(K+1)} (c^{S-k} - c^s)\delta^s(t) - \sum_{j=S-((K+1)-1)}^{(S-k)-1} (c^{S-k} - c^j) \sum_{i=\underline{T}^j}^{\overline{T}^j-1} w(t-i) \quad \forall k \in \{0, \ldots, K-1\}. \tag{10d}$$

As before, (9c) and (9d) just assert the non-negativity of the w variables.

Having projected out $\delta^{S-K}(t)$, we are left with the system

$$v(t) \geq \sum_{s=1}^{S-(K+1)} \delta^s(t) \tag{11a}$$

$$c^{SU}(t) \geq c^{S-k}v(t) - \sum_{s=1}^{S-(K+1)} (c^{S-k} - c^s)\delta^s(t) - \sum_{j=S-((K+1)-1)}^{(S-k)-1} (c^{S-k} - c^j) \sum_{i=\underline{T}^j}^{\overline{T}^j-1} w(t-i) \qquad \forall k \in \{0, \ldots, K\} \tag{11b}$$

$$\delta^s(t) \leq \sum_{i=\underline{T}^s}^{\overline{T}^s-1} w(t-i) \qquad \forall s \in \{1, \ldots, S-(K+1)\} \tag{11c}$$

$$\delta^s(t) \geq 0 \qquad \forall s \in \{1, \ldots, S-(K+1)\} \tag{11d}$$

$$c^{SU}(t) \geq 0, \tag{11e}$$

where (11a) is exactly (10c), and (11b) is (10b) when $k = K$ and (10d) for all other k. Notice (11) is of the form as (7) but with one fewer startup cost indicator, and is exactly (8). □

With Lemma 1, we see that if $S = K + 1$, this is exactly the 3-bin formulation, proving the theorem.

1.2 Match is integer optimal in the binary variables

For this section, we will just consider the space of variables for a single generator, and limit ourselves further by not considering the continuous variables. Consider the uptime/dowtime polytope, and call it Π_{UD}:

$$u(t) - u(t-1) = v(t) - w(t) \qquad \forall t \in \mathcal{T} \tag{12a}$$

$$\sum_{i=t-UT+1}^{t} v(i) \leq u(t) \qquad \forall t \in [UT, T] \tag{12b}$$

$$\sum_{i=t-DT+1}^{t} w(i) \leq 1 - u(t) \qquad \forall t \in [DT, T] \tag{12c}$$

$$u(t), v(t), w(t) \in [0, 1] \qquad \forall t \in \mathcal{T}. \tag{12d}$$

Consider also the uptime/downtime polytope with the Matching variables added on, and call it Π_{UD+M}:

$$
\begin{aligned}
& \text{(12a), (12b), (12c), (12d)} && \\
& \sum_{t'=t-TC+1}^{t-DT} x(t',t) \le v(t) && \forall t \in \mathcal{T} && \text{(13a)} \\
& \sum_{t'=t+DT}^{t+TC-1} x(t,t') \le w(t) && \forall t \in \mathcal{T} && \text{(13b)} \\
& x(t,t') \in [0,1] && \forall t,t' \text{ with } t+DT \le t' \le t+TC-1. && \text{(13c)}
\end{aligned}
$$

We will demonstrate that for any objective over Π_{UD+M} with the property that the startup costs are non-decreasing has an integer optimal solution (when minimizing). Specifically,

$$\min\ c_u^T u + c_v^T v + c_w^T w + c_x^T x \tag{14a}$$

$$\text{s.t. } u, v, w, x \in \Pi_{UD+M} \tag{14b}$$

has integer optimal solutions when:

Assumption 1. $c_x \in \mathbb{R}_-$

Assumption 2. $c_x(t_1,t_2) \le c_x(t_1',t_2')$ *when* $t_2 - t_1 \le t_2' - t_1'$.

The first condition ensures that the cold-start is the most expense start, as $c_x(t_1,t_2) = c^s - c^S$ when $t_2 - t_1 \in [\underline{T}^s, \overline{T}^s)$ ensures $c_x(t_1,t_2) < 0$ (c_v can be adjusted accordingly). The second condition ensures a warmer start is cheaper than a cooler start, and imposes time-consistency among the startup costs (i.e., the startup cost for a particular lag is not dependent on time). These simply encode "startup costs are non-decreasing" into the problem (14). We have the following theorem.

Theorem 1. *Under Assumptions 1 and 2, the optimal vertices of* (14) *are integer.*

In [2, Theorem 7], we showed that optimal solutions to the Match formulation, when startup costs are increasing (i.e., non-decreasing), have the integer decomposition property. Then to prove Theorem 1 we just need to show that optimal solutions that are integer decomposable implies the optimal vertices are integer.

1.2.1 IDP w.r.t. a Set of Objective Vectors Implies Integer Optimal

To review, consider the rational polytope

$$P := \{x \in \mathbb{R}^n \mid Ax \le b\}. \tag{15}$$

We say that the polytope P has the *integer decomposition property* (IDP) if for every positive integer k and for any integer $x \in kP \cap \mathbb{Z}^n$, there exists integer $y_1, \ldots, y_k \in P \cap \mathbb{Z}^n$ such that $x = y_1 + \cdots + y_k$. Baum and Trotter [3] proved that if P has the IDP for every $k \in \mathbb{N}$, then P is an integer polytope. We show their result holds under restriction to a certain set of objective vectors.

Consider a set of vectors $C \subseteq \mathbb{R}^n$. We say that P has the *IDP w.r.t.* C if for every $c \in C$, every $x^* \in \text{argmin}\{c^T x \mid x \in kP \cap \mathbb{Z}^n\}$ is integer decomposable with respect to P. That is, there exists $y_1^*, \ldots, y_k^* \in P \cap \mathbb{Z}^n$ such that $x^* = y_1^* + \cdots + y_k^*$. The following is then immediate.

Lemma 2. *Suppose P has the IDP w.r.t. $C \subseteq \mathbb{R}^n$. Then for every $c \in C$, every optimal extreme point of $\min\{c^T x \mid x \in P\}$ is integer.*

Proof. We follow the proof from [3, Theorem 2]. Let x^* be an optimal extreme point for $\min\{c^T x \mid x \in P\}$, for $c \in C$. As P is rational, we know x^* is rational. Let k be the least common multiple of the components of x^*. Then kx^* is integer. In particular, as x^* was optimal for $\min\{c^T x \mid x \in P\}$, $kx^* \in \text{argmin}\{c^T x \mid x \in kP \cap \mathbb{Z}^n\}$. Since kx^* is integer decomposable with respect to P, there exist $y_1^* + \cdots + y_k^* \in P \cap \mathbb{Z}^n$ such that $kx^* = y_1^* + \ldots + y_k^*$. Hence x^* is a convex combination of the points y_i^* for $i \in \{1, \ldots, k\}$. Since x^* is an extreme point, it must be that $x^* = y_i^*$ for $i \in \{1, \ldots, k\}$. Therefore $x^* \in \mathbb{Z}^n$. □

Lemma 2 shows that if P has the IDP w.r.t. C, then the simplex method will always return an integer solution for $\min\{c^T x \mid x \in P\}$ for every $c \in C$. Put another way, the optimal vertices of $\min\{c^T x \mid x \in P\}$ are integer for every $c \in C$.

In the case of Theorem 1, $C = \{c_u, c_v, c_w, c_x \in \mathbb{R}^{3|\mathcal{T}|+(TC-DT)|\mathcal{T}|} \mid c_x$ satisfies Assumptions 1 and 2$\}$.

1.3 Match and EF have the same LP relaxation values when startup costs are non-decreasing

With the result from Section 1.2, we can say a little more then specifically about the Match formulation when other variables (e.g., those on power) are included. First, we need to specify the EF polytope, Π_{EF}:

$$\sum_{\{t'|t'\geq t+UT\}} y(t,t') = v(t) \qquad \forall t \in \mathcal{T} \tag{16a}$$

$$\sum_{\{t'|t'\leq t-UT\}} y(t',t) = w(t) \qquad \forall t \in \mathcal{T} \tag{16b}$$

$$\sum_{\{t'|t'\leq t-DT\}} x(t',t) = v(t) \qquad \forall t \in \mathcal{T} \tag{16c}$$

$$\sum_{\{t'|t'\geq t+DT\}} x(t,t') = w(t) \qquad \forall t \in \mathcal{T} \tag{16d}$$

$$\sum_{\{\tau,\tau'|\tau\leq t<\tau' \text{ with } \tau'\geq\tau+UT\}} y(\tau,\tau') = u(t) \qquad \forall t \in \mathcal{T} \tag{16e}$$

$$y(t,t'),\ x(t,t') \in [0,1] \qquad \forall t,t', \tag{16f}$$

and call its vertices V_{EF}. As a shortest path formulation, all the vertices V_{EF} are integer [4].

Because the addition of the x variables cuts off no feasible solutions for (12), we know $\text{proj}_{u,v,w}(\Pi_{UD+M}) = \Pi_{UD}$. We also know from above that Π_{UD+M} has integer optimal vertices for vectors $c \in C := \{c_u, c_v, c_w, c_x \in \mathbb{R}^{3|\mathcal{T}|+(TC-DT)|\mathcal{T}|} \mid c_x$ satisfies Assumptions 1 and 2$\}$. Call these vertices the "optimal vertices w.r.t. C", and label them V^C_{UD+M}. Further, let V_{UD} be the vertices of Π_{UD}. We have the following Lemma.

Lemma 3. *For every vertex in V_{UD}, there is exactly one corresponding vertex in V^C_{UD+M}*

Proof. Notice V^C_{UD+M} has only integer vertices by Theorem 1, and C has no restrictions on the values of c_u, c_v, c_w. Take any vertex $\hat{u}, \hat{v}, \hat{w} \in V_{UD}$. The corresponding set of active constraints from Π_{UD} is a cone K, and selecting $-c_u, -c_v, -c_w$ in the polar of K and any c_x satisfying Assumptions 1 and 2 yields a vertex $\hat{u}, \hat{v}, \hat{w}, \hat{x} \in V^C_{UD+M}$ (whose projection is $\hat{u}, \hat{v}, \hat{w}$). Hence $\text{proj}_{u,v,w}(V^C_{UD+M}) = V_{UD}$.

To see the preimage for $\hat{u}, \hat{v}, \hat{w} \in V_{UD}$ under the projection operator is unique, note that for this vertex we can select the greedy solution for the corresponding $\hat{x}$, that is, set $\hat{x}(t_1, t_2) = 1$ if and only if there is a shutdown at t_1 and the very next startup is at t_2, with $t_1 + TC > t_2$. By Assumptions 1 and 2 this choice is a minimizer for $c \in C$, so $\hat{u}, \hat{v}, \hat{w}, \hat{x} \in V^C_{UD+M}$.

To prove the uniqueness suppose we selected a $\hat{x}(t_1, t_2) = 1$ with $t_1 + TC > t_2$, $\hat{w}(t_1) = 1$, $\hat{v}(t_2) = 1$ and t_2 is not the startup immediately following t_1. Hence there is a sooner startup, at time t_2', following the shutdown at t_1. There is another shutdown following it which proceeds t_2, occurring at some time t_1'. Now setting $\hat{x}(t_1, t_2) = 0$ and $\hat{x}(t_1, t_2') = 1$ and $\hat{x}(t_1', t_2) = 1$ results is a strictly better solution under

Assumptions 1 and 2. Hence no such $\hat{x}$ is in V^C_{UD+M}. □

Remark 1. *There exists bijective $f : V_{EF} \to V^C_{UD+M}$ which preserves u, v, w.*

Remark 1 follows by noting the greedy solution given above can be constructed into a full solution for V_{EF} by filling in the remaining arcs. This implies such a f is surjective. Because the vertices of Π_{EF} and Π_{UD} uniquely encode the same set of feasible solutions, we know $|V_{EF}| = |V_{UD}|$. Then we have $|V_{UD}| = |V_{EF}| \geq |V^C_{UD+M}| = |V_{UD}|$, so f must be bijective.

Now we turn to how the work above plays into a full unit commitment formulation, such as (22). For a given LP optimal solution to (22), for each generator consider just its u, v, w variables. The optimal solution u^*, v^*, w^* is in Π_{UD}. Thus it must be a convex combination of (integer) vertices in V_{UD}. Each vertex in V_{UD} has a corresponding vertex in V^C_{UD+M}. Because we have an optimal LP value, the x variables do not appear in any other constraints in (22), and the objective coefficients satisfy Assumptions 1 and 2, x^* must be a convex combination of vertices in V^C_{UD+M}. (If not, we could pivot and select a different set of active constraints corresponding to vertices in V^C_{UD+M}, improve the objective, and not change the current solution as $\text{proj}_{u,v,w}(V^C_{UD+M}) = V_{UD}$.) Finally, by Remark 1 each of these vertices from V^C_{UD+M} have corresponding vertices in V_{EF} with identical objective value. Hence there exists x^{**} with identical objective value which is feasible for $\text{proj}_{u,v,w,x}(\Pi_{EF})$.

Therefore EF and Match have the same LP relaxation values, when startup costs are non-decreasing. Further, this proof shows that any 1-UC formulation including power which has integer optimal vertices in u, v, w also has integer optimal vertices when using Match for non-decreasing startup costs, for example those in [5, 6].

2 Full Specification of the Tested Formulations

In this section we specify the unit commitment formulations tested in [1]. All six formulations share a common set of constraints and variables on generator operation and system balance.

$$\sum_{g\in\mathcal{G}} \left(p_g(t) + \underline{P}_g u_g(t)\right) + p_W(t) = D(t) \qquad \forall t \in \mathcal{T} \tag{17a}$$

$$\sum_{g\in\mathcal{G}} r_g(t) \geq R(t) \qquad \forall t \in \mathcal{T} \tag{17b}$$

$$p_g(t) + r_g(t) \leq (\overline{P}_g - \underline{P}_g)u_g(t) - (\overline{P}_g - SU_g)v_g(t) \qquad \forall t \in \mathcal{T},\ \forall g \in \mathcal{G}^1 \tag{17c}$$

$$p_g(t) + r_g(t) \leq (\overline{P}_g - \underline{P}_g)u_g(t) - (\overline{P}_g - SD_g)w_g(t+1) \qquad \forall t \in \mathcal{T},\ \forall g \in \mathcal{G}^1 \tag{17d}$$

$$p_g(t) + r_g(t) \leq (\overline{P}_g - \underline{P}_g)u_g(t) - (\overline{P}_g - SU_g)v_g(t) - (\overline{P}_g - SD_g)w_g(t+1) \qquad \forall t \in \mathcal{T},\ \forall g \in \mathcal{G}^{>1} \tag{17e}$$

$$p_g(t) + r_g(t) - p_g(t-1) \leq RU_g \qquad \forall t \in \mathcal{T},\ \forall g \in \mathcal{G} \tag{17f}$$

$$p_g(t-1) - p_g(t) \leq RD_g \qquad \forall t \in \mathcal{T},\ \forall g \in \mathcal{G} \tag{17g}$$

$$p_g(t) = \sum_{l\in\mathcal{L}_g} p_g^l(t) \qquad \forall t \in \mathcal{T},\ \forall g \in \mathcal{G} \tag{17h}$$

$$p_g^l(t) \leq (\overline{P}_g^l - \overline{P}_g^{l-1})u_g(t) \qquad \forall t \in \mathcal{T},\ \forall l \in \mathcal{L}_g,\ \forall g \in \mathcal{G} \tag{17i}$$

$$u_g(t) - u_g(t-1) = v_g(t) - w_g(t) \qquad \forall t \in \mathcal{T},\ \forall g \in \mathcal{G} \tag{17j}$$

$$\sum_{i=t-UT_g+1}^{t} v_g(i) \leq u_g(t) \qquad \forall t \in [UT_g, T],\ \forall g \in \mathcal{G} \tag{17k}$$

$$\sum_{i=t-DT_g+1}^{t} w_g(i) \leq 1 - u_g(t) \qquad \forall t \in [DT_g, T],\ \forall g \in \mathcal{G} \tag{17l}$$

$$p_W(t) \leq W(t) \qquad \forall t \in \mathcal{T} \tag{17m}$$

$$p_g^l(t) \in \mathbb{R}_+ \qquad \forall t \in \mathcal{T},\ \forall l \in \mathcal{L}_g,\ \forall g \in \mathcal{G} \tag{17n}$$

$$p_g(t), r_g(t) \in \mathbb{R}_+ \qquad \forall t \in \mathcal{T},\ \forall g \in \mathcal{G} \tag{17o}$$

$$p_W(t) \in \mathbb{R}_+ \qquad \forall t \in \mathcal{T} \tag{17p}$$

$$u_g(t), v_g(t), w_g(t) \in \{0,1\} \qquad \forall t \in \mathcal{T},\ \forall g \in \mathcal{G}. \tag{17q}$$

2.1 One Binary Formulation (1-bin)

$$\min \sum_{g\in\mathcal{G}}\sum_{t\in\mathcal{T}}\left(\sum_{l\in\mathcal{L}_g}(c_g^l\, p_g^l(t)) + c_g^u\, u_g(t) + c_g^{SU}(t)\right) \tag{18a}$$

subject to:

Constraints (17a) – (17q)

$$c_g^{SU}(t) \geq c_g^s\left(u_g(t) - \sum_{i=1}^{\underline{T}_g^s} u_g(t-i)\right) \qquad \forall s\in\mathcal{S}_g,\ \forall g\in\mathcal{G},\ \forall t\in\mathcal{T} \tag{18b}$$

$$c_g^{SU}(t) \geq 0 \qquad \forall g\in\mathcal{G},\ \forall t\in\mathcal{T} \tag{18c}$$

2.2 Strengthened One Binary Formulation (1-Bin*)

$$\min \sum_{g\in\mathcal{G}}\sum_{t\in\mathcal{T}}\left(\sum_{l\in\mathcal{L}_g}(c_g^l\, p_g^l(t)) + c_g^u\, u_g(t) + c_g^{SU}(t)\right) \tag{19a}$$

subject to:

Constraints (17a) – (17q)

$$c_g^{SU}(t) \geq c_g^s\left(u_g(t) - \sum_{i=1}^{DT_g} u_g(t-i)\right) - \sum_{k=1}^{s-1}\left((c_g^s - c_g^k)\sum_{i=\underline{T}_g^k+1}^{\overline{T}_g^k} u_g(t-i)\right) \qquad \forall s\in\mathcal{S}_g,\ \forall g\in\mathcal{G},\ \forall t\in\mathcal{T} \tag{19b}$$

$$c_g^{SU}(t) \geq 0 \qquad \forall g\in\mathcal{G},\ \forall t\in\mathcal{T} \tag{19c}$$

2.3 Three Binary Formulation (3-bin)

$$\min \sum_{g\in\mathcal{G}}\sum_{t\in\mathcal{T}}\left(\sum_{l\in\mathcal{L}_g}(c_g^l\, p_g^l(t)) + c_g^u\, u_g(t) + c_g^{SU}(t)\right) \tag{20a}$$

subject to:

$$
\begin{aligned}
& \text{Constraints (17a) – (17q)} \\
& c_g^{SU}(t) \geq c_g^s v_g(t) \\
& \qquad - \sum_{k=1}^{s-1} \left((c_g^s - c_g^k) \sum_{i=\underline{T}_g^k}^{\overline{T}_g^k - 1} w_g(t-i) \right) \\
& \qquad\qquad\qquad\qquad \forall s \in \mathcal{S}_g,\ \forall g \in \mathcal{G},\ \forall t \in \mathcal{T} && \text{(20b)} \\
& c_g^{SU}(t) \geq 0 \qquad\qquad \forall g \in \mathcal{G},\ \forall t \in \mathcal{T} && \text{(20c)}
\end{aligned}
$$

2.4 Startup Type Indicator Formulation (STI)

$$
\min \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} \left(\sum_{l \in \mathcal{L}_g} (c_g^l\, p_g^l(t)) + c_g^u\, u_g(t) + \sum_{s=1}^{S} c^s \delta^s(t) \right) \tag{21a}
$$

subject to:

$$
\begin{aligned}
& \text{Constraints (17a) – (17q)} \\
& \delta_g^s(t) \leq \sum_{i=\underline{T}_g^s}^{\overline{T}_g^s - 1} w_g(t-i) && \forall s \in \mathcal{S}_g \setminus \{S_g\},\ \forall g \in \mathcal{G},\ \forall t \in \mathcal{T} && \text{(21b)} \\
& v_g(t) = \sum_{s=1}^{S_g} \delta_g^s(t) && \forall g \in \mathcal{G},\ \forall t \in \mathcal{T}. && \text{(21c)}
\end{aligned}
$$

2.5 Matching Formulation (Match)

$$
\begin{aligned}
\min \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} \Bigg(& \sum_{l \in \mathcal{L}_g} (c_g^l\, p_g^l(t)) + c_g^u\, u_g(t) \\
& + c_g^S v_g(t) + \sum_{s=1}^{S_g - 1} (c_g^s - c_g^S) \left(\sum_{t'=t-\overline{T}_g^s+1}^{t-\underline{T}_g^s} x_g(t', t) \right) \Bigg)
\end{aligned} \tag{22a}
$$

subject to:

$$\text{Constraints (17a) – (17q)}$$

$$\sum_{t'=t-TC_g+1}^{t-DT_g} x_g(t',t) \leq v_g(t) \qquad \forall g \in \mathcal{G},\ \forall t \in \mathcal{T} \tag{22b}$$

$$\sum_{t'=t+DT_g}^{t+TC_g-1} x_g(t,t') \leq w_g(t) \qquad \forall g \in \mathcal{G},\ \forall t \in \mathcal{T}, \tag{22c}$$

2.6 Extended Formulation (EF)

$$\min \sum_{g\in\mathcal{G}} \sum_{t\in\mathcal{T}} \left(\sum_{l\in\mathcal{L}_g} (c_g^l\, p_g^l(t)) + c_g^u\, u_g(t) + \sum_{s=1}^{S_g} c_g^s \left(\sum_{t'=t-\overline{T}^s+1}^{t-\underline{T}^s} x_g(t',t) \right) \right) \tag{23a}$$

subject to:

$$\text{Constraints (17a) – (17q)}$$

$$\sum_{\{t'|t'>t\}} y_g(t,t') = v_g(t) \qquad \forall g \in \mathcal{G},\ \forall t \in \mathcal{T} \tag{23b}$$

$$\sum_{\{t'|t'<t\}} y_g(t',t) = w_g(t) \qquad \forall g \in \mathcal{G},\ \forall t \in \mathcal{T} \tag{23c}$$

$$\sum_{\{t'|t'<t\}} x_g(t',t) = v_g(t) \qquad \forall g \in \mathcal{G},\ \forall t \in \mathcal{T} \tag{23d}$$

$$\sum_{\{t'|t'>t\}} x_g(t,t') = w_g(t) \qquad \forall g \in \mathcal{G},\ \forall t \in \mathcal{T} \tag{23e}$$

$$\sum_{\{\tau,\tau'|\tau\leq t<\tau'\}} y_g(\tau,\tau') = u_g(t) \qquad \forall g \in \mathcal{G},\ \forall t \in \mathcal{T}. \tag{23f}$$

3 Computational Results

In this section we present full tables for the computational results reported in [1]. The computational platform used for all experiments is a Dell PowerEdge T620 with two Intel Xeon E5-2670 processors for a total of 16 cores and 32 threads, 256GB of RAM, running the Ubuntu 14.04.5 operating system. The latest major versions of Gurobi (7.0.1) and CPLEX (12.7.1.0) were used when the experiments were conducted.

When referring to a startup cost formulation, we use the same notation as in [1]. That is, "EF" is the extended formulation from [4], "Match" is the matching formulation introduced in [1], "STI" is the startup type indicator formulation introduced in [7], "3-bin" is the three-binary formulation also introduced in [1], "1-bin*" is the strengthened one-binary formulation introduced in [8], and "1-bin" is the typical formulation in the generator's status variables from [9, 10].

We use the same base unit commitment model to benchmark the different startup cost formulations, the full specification of which can be found in the appendix of [1].

3.1 CAISO Instances

We report the computational experiments based on the "CAISO" generators, which are based on real-world market data from the California Independent System Operator. This test set has 610 generators. Four 48-hour demand scenrios are based on historical data corresponding to the date listed (2014-09-01, 2014-12-01, 2015-03-01, 2015-06-01), and one hypothetical high-wind scenario where wind supply is on average 40% of energy demanded (Scenario400).

For each scenario we considered four reserve levels: 0%, 1%, 3%, and 5%. In Tables 1 - 5 for each instance we report the demand/wind scenario followed by the reserve level. A 600 second time limit was imposed for these instances for both solvers.

3.1.1 Gurobi 7.0.1

All Gurobi settings besides the time limit were left at defaults. In Table 1 we report the wall-clock time reported by Gurobi at termination, or if Gurobi hit the 600 second time-limit, we report in parentheses the terminating optimality gap. In the last row we report the geometric mean solve time across the 20 instances for each formulation, inserting 600 seconds into the calculation in the event the solver times out.

As we can see, the EF, 1-bin* and 1-bin variants are uncompetitive. The EF variant is large in comparison to the others, which significantly slows down the initial LP solve as well as root node processing (i.e., heuristics and cut-generation). Conversely, the 1-bin and 1-bin* variants are more compact than Match or STI, but the overall weakness of the formulations (see Table 5) prevents Gurobi from finding and certifying an optimal solution (with $< 0.01\%$ optimality gap) within the time limit. The 3-bin variant is as compact as the 1-bin variants, and while it is more competitive than the latter, for nearly all of these instances it comes in 3rd place behind Match and STI, and fails to solve in one case. The Match and STI variants have broadly similar performance, with Match pulling ahead given its advantage in the hypothetical Scenario400. Based on computational time these instances are "easy" for both Match and STI, in the sense that all 20 instances solve to optimality in under 5 minutes.

In Table 2 we report the number of branch-and-cut nodes explored by Gurobi at termination, indicating with a * when the solver terminated because of the 600 second time limit. In the last row we report the shifted geometric mean node count across all twenty instances (this value is calculated by adding 1 to all node counts and then computing the geometric mean, so as to avoid multiplication by 0).

Table 1: Gurobi Computational Results for CAISO Instances: Wall Clock Time. When instances are solved to optimality, reported quantities are seconds to solution. Otherwise, reported quantities in parentheses are the optimality gap after 600 seconds.

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2014-09-01 0%	357.21	**30.24**	46.80	53.55	(0.028%)	(0.041%)
2014-12-01 0%	169.30	23.89	**23.06**	65.81	(0.073%)	(0.068%)
2015-03-01 0%	166.04	24.68	41.76	**16.46**	(0.042%)	(0.053%)
2015-06-01 0%	163.38	**12.64**	18.76	24.74	(0.017%)	(0.020%)
Scenario400 0%	335.67	**26.60**	65.02	173.37	(0.403%)	(0.383%)
2014-09-01 1%	462.78	**20.39**	22.44	31.83	(0.055%)	(0.045%)
2014-12-01 1%	381.80	36.43	**28.90**	85.78	(0.072%)	(0.069%)
2015-03-01 1%	178.40	**20.41**	35.08	67.20	(0.079%)	(0.090%)
2015-06-01 1%	274.25	41.60	**39.03**	70.83	(0.020%)	(0.028%)
Scenario400 1%	(0.012%)	**46.08**	83.29	182.19	(0.376%)	(0.446%)
2014-09-01 3%	598.73	75.69	**63.26**	87.48	(0.043%)	(0.036%)
2014-12-01 3%	(0.011%)	63.64	**54.88**	93.39	(0.083%)	(0.087%)
2015-03-01 3%	217.10	**48.91**	73.06	99.57	(0.112%)	(0.110%)
2015-06-01 3%	329.79	84.66	**38.13**	83.26	(0.024%)	(0.022%)
Scenario400 3%	(0.013%)	**129.50**	243.01	356.10	(0.495%)	(0.538%)
2014-09-01 5%	412.24	46.80	**44.92**	119.49	(0.037%)	(0.037%)
2014-12-01 5%	(0.012%)	**86.41**	107.14	113.69	(0.104%)	(0.082%)
2015-03-01 5%	(0.010%)	**83.49**	87.22	94.95	(0.115%)	(0.105%)
2015-06-01 5%	(0.010%)	**28.28**	66.97	151.47	(0.031%)	(0.031%)
Scenario400 5%	(0.014%)	115.02	**107.02**	(0.014%)	(0.514%)	(0.570%)
Geometric Mean:	>370.3	**43.12**	52.84	>91.43	>600	>600

As remarked above, because the EF variant is so large, Gurobi only leaves the root node for the EF variant in one instance, and in all other cases Gurobi either finds an optimal solution at the root note or hits the wall-clock limit before beginning to branch. On average Gurobi uses slightly fewer nodes for the Match variant over the STI, and similarly 3-bin, when it solves, only uses a few more nodes on average than STI. Turning to the 1-bin variants, we can see Gurobi processed several thousand nodes in each instance before hitting the time limit, which was not enough to overcome the weakness of these formulations.

3.1.2 CPLEX 12.7.1.0

In Table 3 we report the wall-clock time required by CPLEX to reach an optimal solution, or if the solver hit the time limit, we report the optimality gap at termination in parentheses.

Overall CPLEX performs better on these instances than Gurobi, but we still see the EF and 1-bin variants are uncompetitive. CPLEX is able to solve all the instances using Match, STI, and 3-bin within the time limit, but it is obvious that 3-bin is the inferior of these three: in the worst case (Scenario400 5%) it needs 423 seconds, whereas STI in the worst case needs 223 seconds and Match only needs 110 seconds in the worst case. Though STI outperforms Match in mean solve time, this highlights that Match has

Table 2: Gurobi Computational Results for CAISO Instances: Nodes Explored. Cells report the number of tree nodes explored during branch-and-cut search. Entries with a terminating "*" report the number of tree nodes explored when the 600 second time limit is hit. Otherwise, the entries represent the number of tree nodes required to identify an optimal solution.

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2014-09-01 0%	**0**	**0**	110	**0**	11895*	9285*
2014-12-01 0%	**0**	**0**	**0**	3	5904*	4067*
2015-03-01 0%	**0**	**0**	**0**	**0**	2565*	4419*
2015-06-01 0%	**0**	**0**	**0**	**0**	7247*	15893*
Scenario400 0%	**0**	**0**	**0**	2373	5604*	5801*
2014-09-01 1%	**0**	**0**	**0**	**0**	7164*	10934*
2014-12-01 1%	**0**	95	**0**	160	4072*	7144*
2015-03-01 1%	**0**	**0**	**0**	**0**	2415*	2684*
2015-06-01 1%	**0**	47	47	**0**	11853*	5499*
Scenario400 1%	0*	**0**	47	1854	7210*	6542*
2014-09-01 3%	**0**	7	31	31	14071*	12852*
2014-12-01 3%	0*	1292	366	**144**	3900*	6024*
2015-03-01 3%	**0**	**0**	58	1	2476*	2124*
2015-06-01 3%	**0**	**0**	**0**	**0**	8362*	5153*
Scenario400 3%	0*	**2055**	2874	2309	6518*	6657*
2014-09-01 5%	**0**	95	147	2497	7763*	9316*
2014-12-01 5%	0*	1203	**40**	138	4497*	3681*
2015-03-01 5%	3783*	923	2971	**758**	2264*	2263*
2015-06-01 5%	0*	**0**	100	1125	3894*	5783*
Scenario400 5%	0*	3867	**140**	3848*	6684*	6907*
Shifted Geo. Mean:	>1.510	**13.50**	20.22	>39.09	>5914	>5827

flatter performance profile than STI on this instances. In a similar fashion, we see that 3-bin is sometimes the fastest, but for all the high-wind Scenario400 instances it performs significantly worse than Match or STI. 3-bin underperformed on these instances for Gurobi as well. This is in spite of the fact that STI and 3-bin exhibit the same optimality gap on all these instances (see Table 5). One possible explanation of this phenomenon is the extra indicator variables $\delta_g^s(t)$ make it easier for both Gurobi and CPLEX to generate strong cutting planes. Another possibility is that branching on these indicator variables is often advantageous.

In Table 4 we report the number of branch-and-cut nodes CPLEX explored during search, with a * indicating that the solver terminated because it reached the 600 second wall-clock limit. In the last row we report the shifted geometric mean across the 20 instances, which is calculated the same way it was in Table 2.

We see that for the tighter formulations (EF, Match, STI, and 3-bin), CPLEX often finds and proves an optimal solution at the root node or only a few nodes into the tree. For the 1-bin variants, CPLEX often explores more than 10000 nodes before hitting the wall-clock time limit. Additionally, considering

Table 3: CPLEX Computational Results for CAISO Instances: Wall Clock Time. When instances are solved to optimality, reported quantities are seconds to solution. Otherwise, reported quantities in parentheses are the optimality gap after 600 seconds.

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2014-09-01 0%	277.51	47.36	**32.54**	38.42	(0.060%)	(0.080%)
2014-12-01 0%	176.86	**25.03**	30.41	45.46	(0.103%)	(0.118%)
2015-03-01 0%	167.34	22.82	**19.78**	20.76	(0.082%)	(0.111%)
2015-06-01 0%	141.00	19.68	19.97	**19.01**	(0.023%)	(0.056%)
Scenario400 0%	189.98	**39.40**	52.17	218.87	(0.801%)	(0.956%)
2014-09-01 1%	245.80	46.41	**19.75**	24.74	(0.068%)	(0.097%)
2014-12-01 1%	189.27	41.73	**36.97**	66.30	(0.111%)	(0.146%)
2015-03-01 1%	172.37	37.36	36.69	41.24	(0.059%)	(0.102%)
2015-06-01 1%	188.68	34.94	30.25	**19.89**	(0.026%)	(0.064%)
Scenario400 1%	288.39	63.93	**53.29**	319.76	(0.818%)	(0.874%)
2014-09-01 3%	415.29	62.67	40.45	**37.27**	(0.067%)	(0.108%)
2014-12-01 3%	292.17	72.06	**52.07**	101.31	(0.120%)	(0.135%)
2015-03-01 3%	346.39	58.97	**43.02**	55.51	(0.129%)	(0.102%)
2015-06-01 3%	180.42	38.56	20.41	**19.97**	(0.066%)	(0.073%)
Scenario400 3%	(0.012%)	**110.20**	140.43	420.13	(0.678%)	(0.774%)
2014-09-01 5%	272.24	60.57	**29.44**	53.30	(0.065%)	(0.091%)
2014-12-01 5%	381.81	75.47	**62.67**	102.32	(0.134%)	(0.164%)
2015-03-01 5%	273.29	**35.96**	44.08	58.95	(0.135%)	(0.161%)
2015-06-01 5%	291.17	71.24	**38.51**	59.93	(0.039%)	(0.095%)
Scenario400 5%	(0.012%)	**94.47**	222.68	422.80	(0.766%)	(1.384%)
Geometric Mean:	>261.1	48.02	**40.75**	62.87	>600	>600

instances Scenario400 3% and Scenario400 5%, we observe that Match was able to out-perform STI on these instances because it required less enumeration. Similarly, 3-bin requires more than 5000 nodes on each of the Scenario400 instances, explaining its relative weakness on these high-wind instances.

3.1.3 Relative Integrality Gap

In Table 5 we report the relative integrality gap for each instance and formulation. This is calculated by solving the LP relaxation for each problem and instance, which has value z^*_{LP}, and comparing that to the best integer solution found across all twelve runs for each instance, z^*_{IP}. The corresponding integrality gap can be then calculated by appealing to the formula

$$\text{relative integrality gap} = \frac{z^*_{IP} - z^*_{LP}}{z^*_{IP}}. \tag{24}$$

The values in Table 5 report this ratio as a percentage.

Examining Table 5, we see that EF and Match, as well as STI and 3-bin, always have identical gaps. One way of viewing the observed equivalence of EF and Match is that although Match is not a perfect

Table 4: CPLEX Computational Results for CAISO Instances: Nodes Explored. Cells report the number of tree nodes explored during branch-and-cut search. Entries with a terminating "*" report the number of tree nodes explored when the 600 second time limit is hit. Otherwise, the entries represent the number of tree nodes required to identify an optimal solution.

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2014-09-01 0%	**0**	**0**	**0**	**0**	33839*	31955*
2014-12-01 0%	**0**	**0**	**0**	**0**	15956*	17895*
2015-03-01 0%	**0**	**0**	**0**	**0**	18359*	17284*
2015-06-01 0%	**0**	**0**	**0**	**0**	26114*	23025*
Scenario400 0%	**0**	**0**	**0**	5701	8586*	6955*
2014-09-01 1%	**0**	**0**	**0**	**0**	35138*	28739*
2014-12-01 1%	**0**	**0**	**0**	263	16691*	15893*
2015-03-01 1%	**0**	**0**	**0**	**0**	25618*	23716*
2015-06-01 1%	**0**	**0**	**0**	**0**	24161*	20693*
Scenario400 1%	**0**	**0**	**0**	5748	8766*	7822*
2014-09-01 3%	45	41	126	**0**	24013*	26919*
2014-12-01 3%	2	**0**	21	1531	13099*	14593*
2015-03-01 3%	5	**0**	2	137	16259*	20094*
2015-06-01 3%	**0**	**0**	**0**	**0**	10014*	18983*
Scenario400 3%	803*	**508**	5662	5833	6445*	6008*
2014-09-01 5%	**0**	3	**0**	38	23902*	23423*
2014-12-01 5%	43	34	**2**	2700	7752*	10689*
2015-03-01 5%	**0**	**0**	36	40	14412*	10718*
2015-06-01 5%	3	4	**0**	6	16081*	11000*
Scenario400 5%	1166*	**62**	6378	5815	5922*	5944*
Shifted Geo. Mean:	>3.60	**2.83**	4.75	>32.6	>15442	>15210

formulation for startup costs like EF is, the only vertices that are fractional in Match are sub-optimal – at least for reasonable (i.e., non-decreasing) startup costs. We suspect a similar situation is playing itself out in the comparison between STI and 3-bin. Turning to the 1-bin variants, we see the optimality gap for these is quite large relative to the other formulations tested, which helps to explain their weak computational performance. Additionally, across these instances Match is able to close 40-90% of the integrality gap over STI, which helps explain its performance despite requiring more integer variables.

3.2 FERC Instances

We report the computational experiments based on the "FERC" generators, which are drawn from the RTO Unit Commitment Test System provided by the Federal Energy Regulatory Commission [11], which itself is based on market data gathered from the PJM Interconnection. The FERC set of generators consists of a "Winter" set and a "Summer" set, and each test set has approximately 900 generators. Demand, reserve, and wind scenarios for 2015 were constructed based on market data available on the PJM website [12, 13]. Twelve days were selected from 2015, one from each month, to create a variety of scenarios. We used the

Table 5: Computational Results for CAISO Instances: Relative Integrality Gap (%).

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2014-09-01 0%	0.0097	0.0097	0.0229	0.0229	0.9878	1.0506
2014-12-01 0%	0.0058	0.0058	0.0190	0.0190	1.0813	1.1370
2015-03-01 0%	0.0020	0.0020	0.0270	0.0270	1.5774	1.5774
2015-06-01 0%	0.0012	0.0012	0.0102	0.0102	0.8885	0.8915
Scenario400 0%	0.0113	0.0113	0.1288	0.1288	4.6156	4.6972
2014-09-01 1%	0.0106	0.0106	0.0239	0.0239	1.0058	1.0682
2014-12-01 1%	0.0059	0.0059	0.0198	0.0198	1.0906	1.1509
2015-03-01 1%	0.0037	0.0037	0.0326	0.0326	1.6411	1.6411
2015-06-01 1%	0.0044	0.0044	0.0134	0.0134	0.9105	0.9105
Scenario400 1%	0.0128	0.0128	0.1302	0.1302	4.6721	4.7553
2014-09-01 3%	0.0149	0.0149	0.0283	0.0283	1.0452	1.1093
2014-12-01 3%	0.0089	0.0089	0.0245	0.0245	1.1165	1.1803
2015-03-01 3%	0.0119	0.0119	0.0428	0.0428	1.7416	1.7446
2015-06-01 3%	0.0087	0.0087	0.0180	0.0180	0.9373	0.9451
Scenario400 3%	0.0201	0.0201	0.1372	0.1372	4.7249	4.8072
2014-09-01 5%	0.0081	0.0081	0.0217	0.0217	1.0657	1.1348
2014-12-01 5%	0.0107	0.0107	0.0265	0.0265	1.1415	1.2094
2015-03-01 5%	0.0091	0.0091	0.0459	0.0459	1.7700	1.7798
2015-06-01 5%	0.0084	0.0084	0.0181	0.0181	0.9474	0.9559
Scenario400 5%	0.0237	0.0237	0.1400	0.1400	4.7721	4.8568
Geometric Mean:	0.0079	0.0079	0.0328	0.0328	1.5251	1.5688

Summer generators for the months April – September and the Winter generators for the remaining months.

Using the data collected, we determined wind power was 2% of load, on average, in 2015. We created then for each day selected two scenarios, one with the actual wind data from 2015 (2% Wind Penetration), and another where the wind data from 2015 was multiplied by a constant factor of 15 (30% Wind Penetration). Hence the 2% wind scenarios correspond to the problem facing system operators today, whereas the 30% wind scenarios correspond to problems that system operators may face in the future under high renewables penetration.

For all solvers a time limit of 900 seconds was imposed for these computational experiments.

3.2.1 Gurobi 7.0.1

Because this test set is larger than CAISO, Gurobi often selects the deterministic concurrent optimizer to solve the root LP (this solves the root node using one core for primal simplex, one core for dual simplex, and the remaining cores for parallel barrier). Preliminary experiments showed that this choice resulted in a random, and often large (i.e. greater than 30 seconds), "concurrent spin time," which is the time spend ensuring this concurrent LP solver is deterministic. Gurobi recommended setting the `Method` parameter to 3 to eliminate this lag, which selects the non-deterministic concurrent optimizer. This is the same LP solver

Table 6: Gurobi Computational Results for FERC Instances: Wall Clock Time. When instances are solved to optimality, reported quantities are seconds to solution. Otherwise, reported quantities in parentheses are the optimality gap after 900 seconds.

(a) 2% Wind Penetration

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2015-01-01	511.91	**111.34**	193.07	241.63	(0.017%)	(0.046%)
2015-02-01	586.95	**85.12**	314.07	463.66	(0.143%)	(0.172%)
2015-03-01	807.3	**152.24**	177.44	245.77	649.54	596.24
2015-04-01	(0.012%)	190.62	321.6	**177.27**	675.45	660.92
2015-05-01	512.55	**177.51**	191.29	186.68	334.17	416.03
2015-06-01	619.8	142.57	**139.16**	211.92	406.68	575.42
2015-07-01	(0.017%)	411.00	491.22	**260.41**	(0.014%)	901.87
2015-08-01	808.34	**113.13**	350.52	449.67	(0.11%)	(0.165%)
2015-09-01	(0.016%)	313.79	**284.31**	840.5	(0.101%)	(0.113%)
2015-10-01	605.11	132.95	**113.69**	133.48	582.63	582.58
2015-11-02	573.13	**109.88**	200.83	209.22	(0.073%)	(0.136%)
2015-12-01	(0.013%)	116.25	**114.15**	242.18	(0.055%)	(0.105%)
Geometric Mean:	>701.4	**153.60**	218.36	266.53	>710.7	>738.6

(b) 30% Wind Penetration

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2015-01-01	712.53	**127.22**	(0.902%)	(1.334%)	(4.083%)	(3.808%)
2015-02-01	612.38	**114.78**	(0.043%)	(0.158%)	(0.952%)	(0.959%)
2015-03-01	895.97	647.78	**480.77**	496.35	(0.386%)	(0.460%)
2015-04-01	(0.024%)	**140.82**	236.23	425.71	(0.276%)	(1.054%)
2015-05-01	(0.016%)	**104.62**	119.06	110.24	312.33	337.55
2015-06-01	698.12	222.54	141.18	**110.06**	(0.408%)	(0.101%)
2015-07-01	(0.015%)	**126.98**	346.18	230.15	(0.222%)	(0.105%)
2015-08-01	(0.019%)	395.87	379.42	**227.92**	(0.768%)	(0.870%)
2015-09-01	(0.012%)	**245.73**	780.9	(0.035%)	(0.254%)	(0.256%)
2015-10-01	(0.036%)	439.03	**352.54**	533.72	617.14	607.19
2015-11-02	789.73	**182.40**	618.73	782.18	(0.803%)	(1.065%)
2015-12-01	674.84	**312.67**	361.35	421.08	(0.035%)	(0.035%)
Geometric Mean:	>807.9	**214.70**	>390.3	>400.9	>798.5	>802.6

without the logic to ensure determinism. Hence we set the `Method` parameter to 3 for the FERC experiments on Gurobi. As the solver almost always solved the root LPs in this case using parallel barrier, a practitioner wanting to ensure determinism could set the `Method` parameter to 2 without losing performance.

In Table 6 we report the wall-clock time for the FERC instances, inserting in parentheses the terminating optimality gap when the solver hits the time limit of 900 seconds. (In the 2% wind penetration case, for the 1-bin formulation, instance 2015-07-01, Gurobi found an optimal solution before the solver terminated, so we report the time.)

For the 2% wind instances we observe the 1-bin variants perform better than the CASIO instances, but they are still uncompetitive with Match, STI, and 3-bin variants. The EF is similarly uncompetitive. We see that 3-bin is significantly worse than Match or STI, and for one instance (2015-09-01) takes over 800 seconds to find an optimal solution, whereas STI in the worst case needs only 491 seconds (2015-07-01), and Match in the worst case needs only 411 seconds (2015-07-01). Overall Match outperforms the other variants on these instances using Gurobi.

Turning to the 30% wind instances, we first note that Match is the only variant that solves all 12 instances, and also dominants the other variants in geometric mean solve time. It is interesting to note, turning to a moment to Table 10, that for the 2015-01-01 and 2015-02-01 instances, Match is able to close 95% and 67% of the integrality gap, respectively, over STI. This explains why only Match and EF were able to solve these instances within the time limit. Here again we find again that the EF and 1-bin variants are uncompetitive, and while 3-bin is sometimes the fastest to a solution (e.g. 2015-08-01), it exhibits more performance variability than either STI or Match.

In Table 7 we report the number of branch-and-cut nodes explored at termination; instances when Gurobi terminated because the time limit of 900 seconds was reached are denoted with a *. In the last row we report the shifted geometric mean across the twelve runs of each wind type, which is calculated the same way it was for Table 5.

Across both wind levels it is interesting to note that Gurobi often spends the majority of the time at the root node, first solving the LP and then in cut generation and root-node heuristics. For the largest formulation, EF, Gurobi either finds the optimal at the root node or terminates without having branched. Additionally, the low node count observed in most of the instances for the 1-bin variants reflect this fact as well; Gurobi spends most of the time at the root node attempting to tighten this formulations with cuts. Using the Match variant Gurobi solves nearly all the 2% wind instances at the root node, and only has to explore a significant portion of the tree for a few of the 30% wind instances.

3.2.2 CPLEX 12.7.1.0

For this experiment all CPLEX settings were preserved at default, save setting the 900 second wall-clock time limit.

In Table 8 we report the wall-clock time using CPLEX for the FERC instances, replacing the time with the terminating optimality gap in parentheses when the solver reaches the 900 second time limit without certifying an optimal solution.

Similar to the experience with the CAISO instances, CPLEX overall performs better on this test set than Gurobi. Examining the solver output suggests that one potential reason for this is CPLEX's dual simplex method was usually successful at finding the optimal LP solution in a reasonable amount of time, at least when compared to Gurobi.

Considering the 2% wind instances, we see that Match, STI, and 3-bin variants solve every instance,

Table 7: Gurobi Computational Results for FERC Instances: Nodes Explored. Cells report the number of tree nodes explored during branch-and-cut search. Entries with a terminating "*" report the number of tree nodes explored when the 900 second time limit is hit.

(a) 2% Wind Penetration

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2015-01-01	**0**	**0**	**0**	**0**	1443*	101*
2015-02-01	**0**	**0**	47	976	3864*	3964*
2015-03-01	**0**	**0**	**0**	**0**	1537	1487
2015-04-01	0*	**0**	**0**	**0**	**0**	**0**
2015-05-01	**0**	**0**	**0**	**0**	**0**	**0**
2015-06-01	**0**	**0**	**0**	**0**	**0**	**0**
2015-07-01	0*	**0**	123	47	281*	505
2015-08-01	**0**	**0**	720	1756	1194*	420*
2015-09-01	0*	**46**	425	3543	4087*	4613*
2015-10-01	**0**	**0**	**0**	**0**	**0**	**0**
2015-11-02	**0**	**0**	**0**	**0**	15*	15*
2015-12-01	0*	**0**	**0**	**0**	46*	30*
Shifted Geo. Mean:	>1.00	**1.38**	5.91	9.03	>67.5	>50.8

(b) 30% Wind Penetration

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2015-01-01	**0**	**0**	4440*	7761*	147*	1592*
2015-02-01	**0**	**0**	2067*	2079*	3791*	3387*
2015-03-01	**0**	47	550	47	31*	47*
2015-04-01	0*	**0**	**0**	47	1858*	79*
2015-05-01	0*	**0**	**0**	**0**	**0**	**0**
2015-06-01	**0**	**0**	**0**	**0**	31*	15*
2015-07-01	0*	**0**	**0**	**0**	15*	47*
2015-08-01	0*	879	60	**0**	15*	31*
2015-09-01	0*	**0**	2573	4180*	5759*	4241*
2015-10-01	0*	2501	1775	2199	2100	**1161**
2015-11-02	**0**	**0**	256	390	31*	15*
2015-12-01	**0**	**0**	**0**	387	655*	603*
Shifted Geo. Mean:	>1.00	**4.66**	>51.7	>78.2	>142	>130

with STI exhibiting the best performance overall. Similar to before, the EF, 1-bin*, and 1-bin variants are not competitive. Looking at just Match, STI, and 3-bin, the 3-bin variant exhibits severe performance variability: it solves five of the twelve instances the fastest, but it has the worst-case longest run time of these three – 738 seconds vs. 222 seconds for Match and 150 seconds for STI.

Turning to the 30% wind instances, we note that Match is the only variant able to solve all twelve instances in the time limit required, and is the fastest in geometric mean. Interestingly CPLEX was able to solve the instance 2015-02-01 using STI in a reasonable time. Comparing the terminating optimality gaps, we see that for instance 2015-01-01, Gurobi terminated with a gap of 0.902% for STI, whereas CPLEX terminated with a gap of only 0.078%. This suggests CPLEX may be better than Gurobi at tightening the

Table 8: CPLEX Computational Results for FERC Instances: Wall Clock Time. When instances are solved to optimality, reported quantities are seconds to solution. Otherwise, reported quantities in parentheses are the optimality gap after 900 seconds.

(a) 2% Wind Penetration

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2015-01-01	340.32	127.12	116.39	**103.33**	(0.017%)	(0.016%)
2015-02-01	382.86	**123.08**	144.91	737.72	(0.264%)	(0.261%)
2015-03-01	624.50	127.39	**100.59**	172.34	498.82	578.84
2015-04-01	602.51	**115.71**	144.15	134.50	292.09	297.05
2015-05-01	323.48	83.25	**73.91**	108.68	206.31	153.57
2015-06-01	353.67	119.69	94.44	**79.89**	256.10	339.02
2015-07-01	868.58	221.59	93.79	**87.97**	352.34	516.29
2015-08-01	344.70	131.51	**92.35**	193.95	(0.135%)	(0.130%)
2015-09-01	(0.011%)	143.91	**127.92**	527.99	(0.144%)	(0.148%)
2015-10-01	445.44	164.97	150.02	**143.53**	355.31	393.93
2015-11-02	475.37	175.06	134.81	**129.18**	867.88	(0.017%)
2015-12-01	440.35	138.84	**122.01**	131.57	427.34	523.87
Geometric Mean:	>477.6	135.60	**113.70**	162.42	>498.3	>536.1

(b) 30% Wind Penetration

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2015-01-01	455.13	**155.13**	(0.078%)	(1.252%)	(4.206%)	(4.163%)
2015-02-01	489.58	**165.69**	310.78	(0.152%)	(1.975%)	(1.740%)
2015-03-01	618.62	**179.55**	214.57	242.20	(0.112%)	(0.114%)
2015-04-01	857.72	247.31	258.42	**210.11**	(0.753%)	(0.774%)
2015-05-01	414.45	128.01	101.18	**82.64**	262.59	240.91
2015-06-01	460.15	166.51	109.42	**100.83**	(0.035%)	(0.040%)
2015-07-01	506.11	182.80	131.57	**121.17**	(0.037%)	(0.041%)
2015-08-01	(0.019%)	196.02	161.15	**140.66**	(0.198%)	(0.162%)
2015-09-01	896.64	178.61	**173.09**	736.56	(0.949%)	(0.607%)
2015-10-01	(0.012%)	**269.18**	277.81	559.67	514.99	738.49
2015-11-02	447.73	**189.60**	248.55	(0.022%)	(0.480%)	(0.260%)
2015-12-01	636.79	202.57	**176.64**	215.01	(0.104%)	(0.096%)
Geometric Mean:	>604.1	**185.02**	>210.8	>296.8	>775.3	>793.2

STI formulation either through cuts or presolve, which may explain the difference in performance between the two solvers. For the other variants these instances are largely similar to those preceding: 3-bin exhibits performance variability and is inferior to both Match and STI, and the EF, 1-bin*, and 1-bin variants are uncompetitive.

In Table 9 we report the number of nodes explored at termination, denoting with a * when the solver terminated because it reached the 900 second wall-clock time limit.

Taking both wind levels together, observe for the Match variant CPLEX solves all but one instance at the root node, and for the STI variant it solves all but three of the 24 instances at the root note. In a similar fashion, when the EF variant solves it is often at the root node. The 3-bin variant also solves most of the

Table 9: CPLEX Computational Results for FERC Instances: Nodes Explored. Cells report the number of tree nodes explored during branch-and-cut search. Entries with a terminating "*" report the number of tree nodes explored when the 900 second time limit is hit.

(a) 2% Wind Penetration

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2015-01-01	**0**	**0**	**0**	**0**	5704*	5696*
2015-02-01	**0**	**0**	**0**	5688	3434*	3825*
2015-03-01	11	**0**	**0**	**0**	2274	3868
2015-04-01	**0**	**0**	**0**	**0**	84	88
2015-05-01	**0**	**0**	**0**	**0**	73	**0**
2015-06-01	**0**	**0**	**0**	**0**	**0**	**0**
2015-07-01	**0**	**0**	**0**	**0**	722	1943
2015-08-01	**0**	**0**	**0**	1045	5672*	5694*
2015-09-01	0*	**0**	**0**	5507	5658*	5826*
2015-10-01	**0**	**0**	**0**	**0**	**0**	27
2015-11-02	**0**	**0**	**0**	**0**	3768	2871*
2015-12-01	**0**	**0**	**0**	**0**	165	647
Shifted Geo. Mean:	>1.23	**1.00**	**1.00**	7.52	>355.5	>413.8

(b) 30% Wind Penetration

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2015-01-01	**0**	**0**	2576*	3790*	1094*	1384*
2015-02-01	**0**	**0**	259	5530*	2145*	2370*
2015-03-01	**0**	**0**	**0**	**0**	3005*	2748*
2015-04-01	**0**	**0**	**0**	**0**	1856*	1762*
2015-05-01	**0**	**0**	**0**	**0**	**0**	**0**
2015-06-01	**0**	**0**	**0**	**0**	2335*	2366*
2015-07-01	**0**	**0**	**0**	**0**	3207*	2901*
2015-08-01	0*	**0**	**0**	**0**	1570*	1284*
2015-09-01	**0**	**0**	**0**	5593	3468*	2497*
2015-10-01	3196*	2849	2723	**144**	4438	5830
2015-11-02	**0**	**0**	**0**	3990*	1288*	1608*
2015-12-01	**0**	**0**	**0**	**0**	2164*	1669*
Shifted Geo. Mean:	>1.95	**1.94**	>5.91	>25.3	>1171	>1153

instances at the root node as well. For the 1-bin variants, CPLEX often explores more nodes than Gurobi, but only explores a few thousand before the wall-clock time limit is reached.

3.2.3 Relative Integrality Gap

In Table 10 we report the relative integrality gap for the FERC instances, calculated in the exact same fashion as the CAISO relative integrality gap results reported in Table 5.

First, we observe the same pattern as we did for CAISO: the integrality gaps for EF and Match are always the same, as are those for STI and 3-bin. Otherwise, the results here are significantly different than those for CAISO. We note 1-bin* is no tighter than 1-bin for the FERC instances. Turning to the 2% wind

Table 10: Computational Results for FERC Instances: Relative Integrality Gap (%)

(a) 2% Wind Penetration

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2015-01-01	0.0284	0.0284	0.0362	0.0362	0.5500	0.5500
2015-02-01	0.0423	0.0423	0.0717	0.0717	0.9187	0.9187
2015-03-01	0.0327	0.0327	0.0334	0.0334	0.3727	0.3727
2015-04-01	0.0540	0.0540	0.0540	0.0540	0.5788	0.5788
2015-05-01	0.0456	0.0456	0.0456	0.0456	0.4131	0.4131
2015-06-01	0.0375	0.0375	0.0375	0.0375	1.0321	1.0321
2015-07-01	0.0796	0.0796	0.0796	0.0796	1.3827	1.3827
2015-08-01	0.1233	0.1233	0.1422	0.1422	1.5661	1.5661
2015-09-01	0.5283	0.5283	0.5542	0.5542	1.9062	1.9063
2015-10-01	0.1140	0.1140	0.1141	0.1141	1.0522	1.0522
2015-11-02	0.0760	0.0760	0.0797	0.0797	1.4377	1.4377
2015-12-01	0.0629	0.0629	0.0654	0.0654	1.1305	1.1305
Geometric Mean:	0.0683	0.0683	0.0746	0.0746	0.9113	0.9113

(b) 30% Wind Penetration

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2015-01-01	0.0924	0.0924	1.7525	1.7525	7.4916	7.4916
2015-02-01	0.1703	0.1703	0.5119	0.5119	3.7359	3.7359
2015-03-01	0.0995	0.0995	0.1140	0.1140	1.9207	1.9207
2015-04-01	0.8124	0.8124	0.8476	0.8476	6.8377	6.8377
2015-05-01	0.0729	0.0729	0.0729	0.0729	1.5792	1.5792
2015-06-01	0.0807	0.0807	0.0859	0.0859	2.5388	2.5388
2015-07-01	0.1383	0.1383	0.1412	0.1412	2.5278	2.5278
2015-08-01	0.3701	0.3701	0.3904	0.3904	4.1100	4.1100
2015-09-01	0.2884	0.2884	0.3747	0.3747	3.2275	3.2275
2015-10-01	1.1342	1.1342	1.1446	1.1446	2.9962	2.9962
2015-11-02	0.1825	0.1825	0.2626	0.2626	2.8615	2.8615
2015-12-01	0.2558	0.2558	0.2738	0.2738	1.2704	1.2704
Geometric Mean:	0.2060	0.2060	0.3141	0.3141	3.0031	3.0031

instances, we see that EF and Match often are not tighter than STI and 3-bin, or are only marginally so. This explains STI's performance dominance on the 2% instances in CPLEX – the extra variables from Match are not, in these instances, buying much (or any) additional tightness over STI. Match and EF only close 8% of the optimally gap in geometric mean over STI, which is significantly less than the 75% geometric mean gap closure observed for CAISO.

Considering now the 30% wind instances, we see in particular that Match closes a large portion of the optimality gap over STI in the 2015-01-01 and 2015-02-01 instances, with modest reductions in every instance except 2015-05-01. We also observe that in general, the high-wind instances, both here and in Table 5, have larger integrality gaps than low-wind instances across all formulations. This should be expected as large amounts of renewables generation imply large net-load swings, which should result in more generator switching and generator ramping.

4 Statistical Analysis

In this section we report the results of a statical analysis of the computational results above, using the Wilcoxon signed-rank test [14]. To separate out the potential contributions to performance variability, we considered five sets of instances for each solver: (i) "All" ($n = 44$) – which consists of the entire test suite, (ii) "CAISO" ($n = 20$) – which is the CAISO set of instances, (iii) "FERC" ($n = 24$) – which is the FERC set of instances, (iv) "High Wind" ($n = 16$) – which consists of the Scenario400 instances from CAISO and the 30% Wind Penetration instances from FERC, and (v) "Low Wind" ($n = 28$) – which is all the other instances not in High Wind. We note that for $n \lessapprox 20$ this statistical test starts to become underpowered.

4.1 Gurobi 7.0.1

In Table 11 we report the mean differences in solve times and the results of the Wilcoxon signed-rank test across the five sets described above on the Gurobi computational experiments. In each cell we report the column mean solve time minus the row mean solve time; hence a negative number implies the column was faster than the row, whereas a positive number implies the row was faster than the column. Because the Wilcoxon test is for difference in arithmetic mean, the results in these tables report the difference in arithmetic mean solve time, whereas the summary results in Section 3 report the geometric mean solve time. Looking at the entire test set we can see that the Match formulation outperforms the others at the $\alpha = 0.01$ using Gurobi. Match also outperforms STI in the breakdowns at the $\alpha = 0.05$ level, except for the CAISO test set. STI in turn outperforms 3-bin overall and in several of the breakout sets. These statistics also bear out the larger observation that the EF, 1-bin, and 1-bin* variants are uncompetitive with any of Match, STI, and 3-bin.

4.2 CPLEX 12.7.1.0

In Table 12 we report the mean differences in solve time and the results of the Wilcoxon signed-rank test for the CPLEX computational experiments. As with Table 11, In each cell we report the column mean solve time minus the row mean solve time; so a negative number implies the column was faster than the row, and a positive number implies the row was faster than the column. While Match still has the best mean overall, the Wilcoxon test is not able to differentiate it from STI and 3-bin. Interestingly STI is better than 3-bin at the $\alpha = 0.01$ level. We also note that on the Low Wind instances STI is able to out-perform Match using CPLEX at the $\alpha = 0.01$, which bears out the observations from the computational results above. The magnitude of the difference is not large, however. Turning to the High Wind instances, we see Match is able

Table 11: Results of the Wilcoxon signed-rank test for Gurobi computational experiments. Each cell reports the column mean solve time minus the row mean solve time. A "*" indicates the difference is significant at the $\alpha = 0.05$ level; a "**" indicates the difference is significant at the $\alpha = 0.01$ level.

(a) All ($n = 44$)

Formulation	EF	Match	STI	3bin	1bin*	1bin
EF		-466.0**	-383.3**	-327.1**	96.5**	101.3**
Match	466.0**		82.7**	138.9**	562.5**	567.3**
STI	383.3**	-82.7**		56.2**	479.8**	484.6**
3bin	327.1**	-138.9**	-56.2**		423.6**	428.4**
1bin*	-96.5**	-562.5**	-479.8**	-423.6**		4.8
1bin	-101.3**	-567.3**	-484.6**	-428.4**	-4.8	

(b) CAISO ($n = 20$)

Formulation	EF	Match	STI	3bin	1bin*	1bin
EF		-360.3**	-348.1**	-284.0**	188.0**	188.5**
Match	360.3**		12.2	76.3**	548.3**	548.8**
STI	348.1**	-12.2		64.1**	536.0**	536.5**
3bin	284.0**	-76.3**	-64.1**		472.0**	472.5**
1bin*	-188.0**	-548.3**	-536.0**	-472.0**		0.5
1bin	-188.5**	-548.8**	-536.5**	-472.5**	-0.5	

(c) FERC ($n = 24$)

Formulation	EF	Match	STI	3bin	1bin*	1bin
EF		-554.1**	-412.7**	-363.0**	20.3	28.6
Match	554.1**		141.5*	191.1*	574.4**	582.8**
STI	412.7**	-141.5*		49.7	432.9**	441.3**
3bin	363.0**	-191.1*	-49.7		383.3**	391.7**
1bin*	-20.3	-574.4**	-432.9**	-383.3**		8.4
1bin	-28.6	-582.8**	-441.3**	-391.7**	-8.4	

(d) High Wind ($n = 16$)

Formulation	EF	Match	STI	3bin	1bin*	1bin
EF		-534.2**	-362.9**	-285.8**	26.0	27.4
Match	534.2**		171.3*	248.5*	560.2**	561.6**
STI	362.9**	-171.3*		77.1	388.9**	390.3**
3bin	285.8**	-248.5*	-77.1		311.7**	313.1**
1bin*	-26.0	-560.2**	-388.9**	-311.7**		1.4
1bin	-27.4	-561.6**	-390.3**	-313.1**	-1.4	

(e) Low Wind ($n = 28$)

Formulation	EF	Match	STI	3bin	1bin*	1bin
EF		-427.1**	-395.0**	-350.7**	136.8**	143.5**
Match	427.1**		32.1*	76.3**	563.8**	570.6**
STI	395.0**	-32.1*		44.3**	531.8**	538.5**
3bin	350.7**	-76.3**	-44.3**		487.5**	494.2**
1bin*	-136.8**	-563.8**	-531.8**	-487.5**		6.7
1bin	-143.5**	-570.6**	-538.5**	-494.2**	-6.7	

Table 12: Results of the Wilcoxon signed-rank test for CPLEX computational experiments. Each cell reports the column mean solve time minus the row mean solve time. A "*" indicates the difference is significant at the $\alpha = 0.05$ level; a "**" indicates the difference is significant at the $\alpha = 0.01$ level.

(a) All ($n = 44$)

Formulation	EF	Match	STI	3bin	1bin*	1bin
EF		-328.3**	-317.1**	-218.3**	212.8**	227.1**
Match	328.3**		11.3	110.0	541.2**	555.4**
STI	317.1**	-11.3		98.8**	529.9**	544.1**
3bin	218.3**	-110.0	-98.8**		431.1**	445.4**
1bin*	-212.8**	-541.2**	-529.9**	-431.1**		14.2
1bin	-227.1**	-555.4**	-544.1**	-445.4**	-14.2	

(b) CAISO ($n = 20$)

Formulation	EF	Match	STI	3bin	1bin*	1bin
EF		-234.5**	-236.2**	-180.1**	314.7**	314.7**
Match	234.5**		-1.7	54.4	549.2**	549.2**
STI	236.2**	1.7		56.0**	550.8**	550.8**
3bin	180.1**	-54.4	-56.0**		494.8**	494.8**
1bin*	-314.7**	-549.2**	-550.8**	-494.8**		0.0
1bin	-314.7**	-549.2**	-550.8**	-494.8**	0.0	

(c) FERC ($n = 24$)

Formulation	EF	Match	STI	3bin	1bin*	1bin
EF		-406.5**	-384.5**	-250.1*	128.0	154.1*
Match	406.5**		22.0	156.4	534.5**	560.6**
STI	384.5**	-22.0		134.4	512.5**	538.6**
3bin	250.1*	-156.4	-134.4		378.1**	404.2**
1bin*	-128.0	-534.5**	-512.5**	-378.1**		26.1
1bin	-154.1*	-560.6**	-538.6**	-404.2**	-26.1	

(d) High Wind ($n = 16$)

Formulation	EF	Match	STI	3bin	1bin*	1bin
EF		-422.2**	-361.8**	-175.4	186.1*	195.3*
Match	422.2**		60.4	246.8*	608.3**	617.5**
STI	361.8**	-60.4		186.4*	547.9**	557.1**
3bin	175.4	-246.8*	-186.4*		361.5**	370.7**
1bin*	-186.1*	-608.3**	-547.9**	-361.5**		9.2
1bin	-195.3*	-617.5**	-557.1**	-370.7**	-9.2	

(e) Low Wind ($n = 28$)

Formulation	EF	Match	STI	3bin	1bin*	1bin
EF		-274.7**	-291.5**	-242.8**	228.1**	245.2**
Match	274.7**		-16.8**	31.9	502.8**	519.9**
STI	291.5**	16.8**		48.7**	519.6**	536.7**
3bin	242.8**	-31.9	-48.7**		470.9**	488.0**
1bin*	-228.1**	-502.8**	-519.6**	-470.9**		17.1*
1bin	-245.2**	-519.9**	-536.7**	-488.0**	-17.1*	

to out-perform the other formulations save STI at the $\alpha = 0.05$ level; it is likely the low power of the test at $n = 16$ makes it difficult to distinguish Match and STI statistically. Finally we observe that overall the EF, 1-bin, and 1-bin* variants are significantly worse than the Match, STI, and 3-bin variants.

5 Summary

Considering Tables 1 and 6 together, it is unambiguous that Match performs better than the other variants on Gurobi, followed by STI and then 3-bin. This is born out in the statical analysis of these results in Table 11. Given that Match is as tight as EF in all instances while needing many fewer variables, but not too many additional variables over STI, this result is not surprising.

Conversely, the computational results using CPLEX reported in Tables 3 and 8 are a bit more ambiguous, and this is reflected in Table 12. While using CPLEX Match is often slower in the average case than STI, using the Match formulation CPLEX solved every of the 44 instances considered in under 5 minutes, and hence it exhibited the better worst-case performance.

References

[1] B. Knueven, J. Ostrowski, and J.-P. Watson, "A novel matching formulation for startup costs in unit commitment," 2018.

[2] ——, "Online companion for exploiting identical generators in unit commitment," 2017. [Online]. Available: https://github.com/jwatsonnm/uc-benchmarking/raw/master/papers/OnlineCompanionExploitingIdentical.pdf

[3] S. Baum and L. E. Trotter Jr, "Integer rounding and polyhedral decomposition for totally unimodular systems," in *Optimization and Operations Research*. Springer, 1978, pp. 15–23.

[4] Y. Pochet and L. A. Wolsey, *Production planning by mixed integer programming*. Springer Science & Business Media, 2006.

[5] C. Gentile, G. Morales-Espana, and A. Ramos, "A tight MIP formulation of the unit commitment problem with start-up and shut-down constraints," *EURO Journal on Computational Optimization*, pp. 1–25, 2016.

[6] B. Knueven, J. Ostrowski, and J. Wang, "The ramping polytope and cut generation for the unit commitment problem," *INFORMS Journal on Computing*, 2017, to appear.

[7] G. Morales-España, J. M. Latorre, and A. Ramos, "Tight and compact MILP formulation for the thermal unit commitment problem," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4897–4908, 2013.

[8] M. Silbernagl, M. Huber, and R. Brandenberg, "Improving accuracy and efficiency of start-up cost formulations in MIP unit commitment by modeling power plant temperatures," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2578–2586, 2016.

[9] M. P. Nowak and W. Römisch, "Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty," *Annals of Operations Research*, vol. 100, no. 1-4, pp. 251–272, 2000.

[10] M. Carrion and J. M. Arroyo, "A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem," *IEEE Transactions on Power Systems*, vol. 21, no. 3, pp. 1371–1378, Aug 2006.

[11] E. Krall, M. Higgins, and R. P. O'Neill, "RTO unit commitment test system," *Federal Energy Regulatory Commission*, 2012.

[12] PJM. PJM - ancillary services. [Online]. Available: http://pjm.com/markets-and-operations/ancillary-services.aspx

[13] ——. PJM - system operations. [Online]. Available: http://www.pjm.com/markets-and-operations/ops-analysis.aspx

[14] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.