12-2022

Novel Mixed Integer Programming Approaches to Unit Commitment and Tool Switching Problems

Najmaddin Akhundov
nakhundo@utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Najmaddin Akhundov entitled "Novel Mixed Integer Programming Approaches to Unit Commitment and Tool Switching Problems." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

James Ostrowski, Major Professor

We have read this dissertation and recommend its acceptance:

Hugh Medal, Mingzhou Jin, Paolo Letizia

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Najmaddin Akhundov entitled "Novel Mixed Integer Programming Approaches to Unit Commitment and Tool Switching Problems." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

James Ostrowski, Major Professor

We have read this dissertation
and recommend its acceptance:

Hugh Medal

Mingzhou Jin

Paolo Letizia

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Novel Mixed Integer Programming Approaches to Unit Commitment and Tool Switching Problems

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Najmaddin Akhundov

December 2022

Acknowledgments

I would like to thank my advisor, Professor Jim Ostrowski, for his guidance during my time as a graduate student. His guidance has played an important role in this dissertation and my professional development. I would also like to thank the members of my defense committee, Professors Paolo Letizia, Mingzhou Jin, and Hugh Medal for their continuous help and support.

I also want to express my deepest gratitude for all the teachers I had in my life time. They made me the man I am today. I devote this thesis to my first teachers, my mother Tarana Gurbanova and my father Nizamaddin Akhundov.

Abstract

In the first two chapters, we discuss mixed integer programming formulations in Unit Commitment Problem. First, we present a new reformulation to capture the uncertainty associated with renewable energy. Then, the symmetrical property of UC is exploited to develop new methods to improve the computational time by reducing redundancy in the search space. In the third chapter, we focus on the Tool Switching and Sequencing Problem. Similar to UC, we analyze its symmetrical nature and present a new reformulation and symmetry-breaking cuts which lead to a significant improvement in the solution time.

In chapter one, we use convex hull pricing to explicitly price the risk associated with uncertainty in large power systems scheduling problems. The uncertainty associated with renewable generation (e.g. solar and wind) is highlighting the need for changes in how power production is scheduled. It is known that symmetry in the integer programming formulations can slow down the solution process due to the redundancy in the search space caused by permutations. In the second chapter, we show that having symmetry in the unit commitment problem caused by having identical generating units could lead to a computational burden even for a small scale problem. We present an effective method to exploit symmetry in the formulation introduced by identical (often co-located) generators. We propose a cut-generation approach coupled with aggregation method to remove symmetry without sacrificing feasibility or optimality. In the third chapter, we focus on the Job Sequencing and Tool Switching Problem (SSP), which is a well-known combinatorial optimization

problem in the domain of Flexible Manufacturing Systems (FMS). We propose a new integer linear programming approach with symmetry-breaking and tightening cuts that provably outperformed the existing methodology described in the literature.

Table of Contents

List of Tables

List of Figures

Chapter 1

Convex Hull Pricing as a Risk Mitigation Device in Unit Commitment

1.1 Introduction

While the increase in renewable generation has helped produce cleaner and cheaper electricity, the increased uncertainty that comes along with it has complicated operations. Although decision making under uncertainty is a well developed branch of operations research, stochastic variants of problems are typically much harder to solve than their deterministic version. This poses a problem in many power systems applications as actionable solutions to large-scale problem are needed quickly. Further, there are market-design issues with implementing stochastic models. Markets based on scenario-based solutions schedule generators before prices are known, something that is unpalatable to generator owners.

Scheduling power generators is determined by solving unit commitment (UC) problems; minimizing the total cost of production while guaranteeing that each generator is given a feasible production schedule and that all demand is met.

In conventional deterministic UC models, due to relatively low marginal costs of renewables, optimal schedules tend to use as much renewable energy as possible. However, these models ignore the cost associated with uncertainty, and the true cost of these schedules might not be accurately accounted for in the model. If the penetration of renewable energy is less than some threshold, the system can manage it at a negligible cost, just as it handles the unavoidable variability in the load. Beyond modest penetration levels, however, uncertainty must be characterized and risk should be quantified to be able to integrate renewables into the mix of power resources while maintaining reliability.

Existing risk management solutions to this challenge can be classified into the following categories: market design, energy storage systems, and UC model building. Currently, electricity market structures in the U.S. are built around the two-settlement concept, which refers to the day-ahead and the real-time markets. Some studies suggest the introduction of intra-day commitments in the U.S. energy markets as renewable forecast uncertainty is significantly lower during intra-day markets than in the day-ahead market Herrero et al., 2018; Tesfatsion, 2020. Grid-scale energy storage technologies are widely believed to have the potential to increase power system flexibility and to effectively integrate high shares of renewables Castillo and Gayme, 2014; Denholm and Hand, 2011; Zhang et al., 2016. However, storage integration is still an ongoing challenge due to the high cost, low energy density, and complex maintenance.

A variety of modeling approaches and solution strategies are proposed and applied to UC problems for assessing the impacts of renewable integration and mitigating the associated uncertainty. Robust, chance-constrained and Stochastic optimization models are among the main approaches van Ackooij et al., 2018. Robust models are based on worst-case outcomes and provide robust commitment schedules for controllable generators (e.g., thermal generators) Jiang et al., 2011; Lorca et al., 2016. Nevertheless, these models tend to be overly conservative and may not be cost-effective An and Zeng, 2014. In chance-constrained optimization models some constraints (e.g., load constraints) must be satisfied with high probability Bertsimas et al., 2012; Wang et al., 2011. However, these models are still largely intractable as the probabilistic constraints may be non-convex and hard to evaluate. Stochastic UC models are considered an important family of approaches as they represent theoretical ideals among UC models and lead to cost savings and reliability improvement. The uncertainty in these models is captured by a set of discrete scenarios with associated probabilities which is usually large and leads to computational intractability for solving large-scale problems. As a result scenario selection strategies and decomposition methods (e.g., Benders' decomposition, column generation) have been subjects of intense research over the last two decades Papavasiliou and Oren, 2013; Schulze and McKinnon, 2016; Tahanan et al., 2015; van Ackooij et al., 2018; Zhao and Guan, 2015; Zheng et al., 2014, however, implementing these methods might require a redesign of current markets.

The focus of this study is building a UC model that captures and mitigates the uncertainty of wind integration in power supply. We provide a methodology that explicitly quantifies the price of the risk using the concept of convex hull pricing (CHP) Gribik et al., 2007, a pricing scheme that is based on the optimality of dual variables associated with system-wide constraints (e.g, demand and transmission constraints). CHP has been proposed in the context of the electricity market clearing process in order to increase market transparency. One desirable property in CHP is that it minimizes uplift payments Hua and Baldick, 2016; Schiro et al., 2015. Due to non-convexities in the energy market that arise from operating characteristics, generators may fail to recover their cost by selling energy at the given prices. In this case, the independent system operators (ISOs) makes side payments, or uplift payments, to participants to compensate them for their loss, encouraging them to follow the ISO's solution. Among the various pricing schemes, CHP represents a theoretical ideal that supports efficient dispatch given the underlying non-convexities Chao, 2019.

We employ CHP to price the uncertainty associated with the variable wind and include this price in the wind generator's conventional operating costs. Performance of the proposed model is evaluated by comparing it with stochastic and deterministic UC models in terms of production cost and solution time. It is shown that the proposed model outperforms the deterministic UC model and we gain the advantage of the stochastic model, with modest additional computational efforts. As mentioned, the stochastic model is an ideal cost-effective model, however, solving it requires significant computational costs in practice. The efficiency of the proposed model is examined with various wind penetration levels.

1.2 Model Formulation

In this section a UC model is presented that takes into account the uncertainty of wind energy in power supply and provides a risk-aware optimal schedule of generation units in the day-ahead market. The uncertainty is formulated using the concept of convex hull pricing (CHP) and probabilistic distribution of wind power forecast.

1.2.1 A Primal Formulation for CHP

Consider the following deterministic mixed-integer linear UC problem:

$$z = \min \sum_{g \in \mathcal{G}} c^g(u^g, p^g) \tag{1.1a}$$

$$\text{s.t.} \ \sum_{g \in \mathcal{G}} (A^g p^g + B^g u^g) + w = D \tag{1.1b}$$

$$(u^g, p^g, c^g) \in \Pi^g, \ \forall g \in \mathcal{G} \tag{1.1c}$$

where u^g, p^g and c^g denote the binary commitment vector, the continuous dispatch vector and the cost vector of the schedule associated with u^g and p^g for generator $g \in \mathcal{G}$, respectively. Matrices A^g and B^g are the corresponding coefficients of the variables, w denotes wind energy which is aggregated and represented as a single resource, and demand D is met by a mixture of resources per constraint (1.1b).

The set Π^g represents the feasible schedules, dispatches and costs for each generator g. These usually encode physical operating constraints like minimum and maximum power outputs, minimum up and down times, and ramping constraints. The objective (1.1a) minimizes the system operational costs of the conventional power generators. Note that in this formulation the cost associated to the wind energy generation is neglected.

Convex hull prices are determined through the Lagrangian relaxation of problem (1.1) with respect to the supply-demand constraint (1.1b):

$$L(\pi(w)) = \min \sum_{g \in \mathcal{G}} c^g + \pi(w)^T \left(D - A^g p^g - B^g u^g - w\right)$$

subject to

$$(u^g, p^g, c^g) \in \Pi^g, \ \forall g \in \mathcal{G}.$$

Note that the dual vector $\pi(w)$ is a function of w, the amount of wind contribution to the system. The associated Lagrangian dual is:

$$\max_{\pi} L(\pi(w)). \tag{1.2}$$

By definition Gribik et al., 2007, convex hull prices are the optimal solutions $\pi(w)$ to the problem (1.2). Alternatively, as shown in Hua and Baldick, 2016, convex hull prices can be calculated as the dual optimal vector with respect to the supply-demand constraint (1.3b) of the following linear program:

$$z(w) = \min \sum_{g \in \mathcal{G}} c^g(u^g, p^g) \tag{1.3a}$$

$$\text{s.t.} \ \sum_{g \in \mathcal{G}} A^g p^g + B^g u^g + w = D \qquad (\pi^{CH}(w)) \tag{1.3b}$$

$$(u^g, p^g, c^g) \in \text{conv}\left(\Pi^g\right), \ \forall g \in \mathcal{G} \tag{1.3c}$$

where conv(.) denotes the convex hull of a given set and $\pi^{CH}(w)$ denotes the CHP vector (Figure 1.1). The objective function, $z(w)$ is a non-increasing function in terms of w and tracks the changes in the solution of the linear relaxation of problem (1.1) when the wind generator contributes w unit to the system. It is known that $z(w)$ is piece-wise linear and its subgradient at w contains $\pi^{CH}(w)$ Bertsimas and Tsitsiklis, 1997.

One potential drawback of using problem (1.3) to obtain the convex hull prices is that it requires explicit knowledge of the convex hull for each generator and constructing it may be computationally challenging in the presence of ramping constraints. Knueven et al., 2019 propose a Benders' decomposition approach to address this problem and demonstrate across a large set of test instances that their decomposition approach only requires modest computational effort. As shown in the same study, the prices can be obtained using "approximated" convex hull prices (aCHP) which can be computed by solving problem (1.3) with the linear relaxation of the set Π^g in constraint (1.3c).

1.2.2 Wind Price Formation

Discrepancies between scheduled and actual wind production must be met by other resources. CHP gives an accurate reflection of the cost of these resources, without strictly identifying which resource will provide for the shortfall. If the wind production is ϵ less than scheduled, for small ϵ, the cost to the system can be approximated as $\epsilon\pi(w_{lp})$. However, at larger deviations, the CHP might not be an accurate prediction for the true cost. In this case, for a better approximation, an integration over the deviation is needed. This motivates the following approximation to the true cost of deviation when w_{lp} units of wind are scheduled:

$$c(w_{lp}) = \int_0^{w_{lp}} \pi^{CH}(w_{lp} - \epsilon) Pr(w \leq w_{lp} - \epsilon) d\epsilon. \tag{1.4}$$

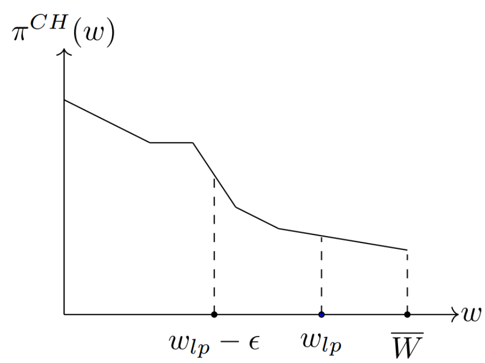

Figure 1.1: CHP as a function of wind in a fixed time period

In this formulation, $c(w_{lp})$ represents the expected cost of wind uncertainty when the wind realization is below w_{lp} and $Pr(\cdot)$ denotes the probability distribution of wind production.

Note that the function $c(w_{lp})$ relies only on the probability distribution of wind and the CHP, not the actual solution obtained for the UC instance. This cost function is added to the objective function (1.1a) in order to reflect the cost of wind uncertainty. Therefore, the proposed unit commitment (PUC) model reads as follows:

$$z^{PUC} = \min \sum_{g \in \mathcal{G}} c^g(u^g, p^g) + c(w) \tag{1.5a}$$

$$\text{s.t.} \sum_{g \in \mathcal{G}} A^g p^g + B^g u^g + w = D \tag{1.5b}$$

$$(u^g, p^g, c^g) \in \Pi^g, \ \forall g \in \mathcal{G}. \tag{1.5c}$$

In the above formulation, the objective function aims to minimize the total generating cost and the penalty of the risks due to the uncertainty of wind power.

A computationally tractable wind cost function is built based on its abstract formulation. The probability density of the wind generation is assumed to follow a uniform distribution. Please note that, if the wind generation follows a different distribution (i.e. Weibull), it could be divided into smaller section that approximately follow a uniform distribution, then the procedure described below would be valid for this case as well. For a fixed single-hour time period $\bar{t}$, various amounts of available wind energy (e.g. 0 to 10% of the demand) is considered and for the remaining time periods $t \neq \bar{t}$, the wind energy is set to its maximum capacity $\overline{W}$. The wind prices, $\pi^{CH}(w)$, is obtained by solving problem (1.3) for the interval $[0, \overline{W}]$ and cluster the prices using a K-means algorithm MacQueen et al., 1967. The wind cost function is computed from equation (1.4). This is a quadratic function in terms of w_{lp}, which will be linearized before including it in the UC problem. Please see the following example.

Example 1. *Figure (1.2) represents the CHPs obtained from equation (1.3b) for a fixed single-time period. By clustering the prices into three pieces with associated sub-intervals* $[0, q_1], [q_1, q_2]$ *and* $[q_2, \overline{W}]$*, we obtain the constant prices* π_1^{CH}*,* π_2^{CH} *and* π_3^{CH} *for each respective interval (Figure 1.3). Now when* $w_{lp} \geq q_2$ *the wind cost function calculates as follows:*

$$c(w_{lp}) = \int_0^{q_1} \pi_1^{CH} \frac{w_{lp} - \epsilon}{\overline{W}} d\epsilon + \int_{q_1}^{q_2} \pi_2^{CH} \frac{w_{lp} - \epsilon}{\overline{W}} d\epsilon + \int_{q_2}^{w_{lp}} \pi_3^{CH} \frac{w_{lp} - \epsilon}{\overline{W}} d\epsilon. \tag{1.6}$$

The obtained wind cost function (Figure 1.4) is non-linear, therefore we piece-wise linearize it prior including it in the objective function of the UC problem.

Figure 1.5 represents linearization by partitioning the interval $[0, \overline{W}]$ *into ten equal sub-intervals and* $[b_i, b_{i+1}]$ *with* $i \in \{0, ..., 9\}$*. Introducing two new set of decision variables* z_i*,* $i \in \{0, ..., 10\}$ *and* $y_j \in \{0, 1\}$*,* $j \in \{0, ..., 10\}$ *we can find the piece-wise approximation of the wind cost function with as following:*

$$c(\cdot) = \sum_i z_i c(b_i) \tag{1.7a}$$

$$w = \sum_i z_i b_i \tag{1.7b}$$

$$z_i \leq y_i \ \forall i \in \{0, ..., 10\} \tag{1.7c}$$

$$z_{i+1} \leq y_i + y_{i+1} \ \forall i \in \{0, ..., 10\} \tag{1.7d}$$

$$\sum_j y_j = 1 \tag{1.7e}$$

$$\sum_i z_i = 1 \tag{1.7f}$$

Here, in equation 1.7a and 1.7b the wind cost function and utilized wind energy amount (w_t) *were set to their approximated values, while equations 1.7d - 1.7f were used to force the model to pick proper linear combination of the calculated cost values* $c(b_i)$*.*

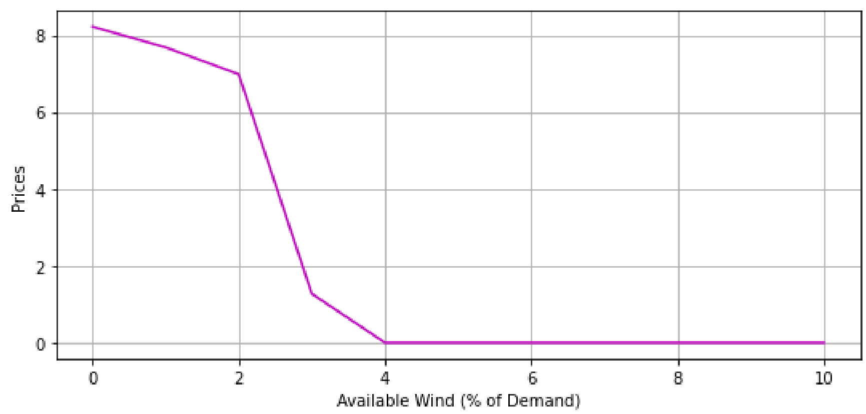

Figure 1.2: Prices of wind energy.

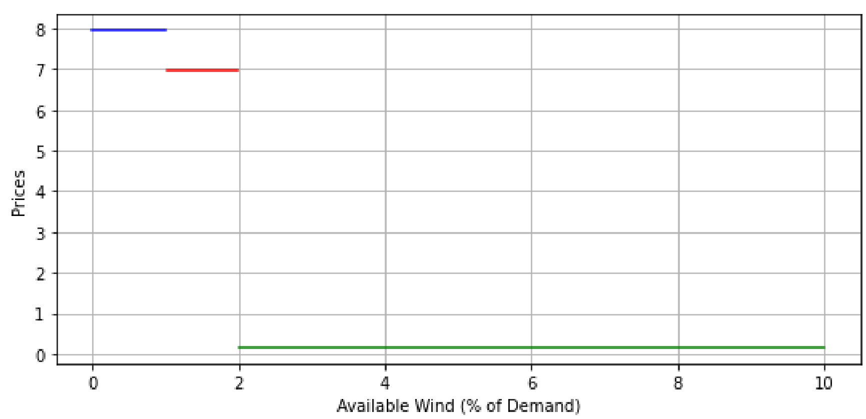

Figure 1.3: Clustered prices of wind energy.

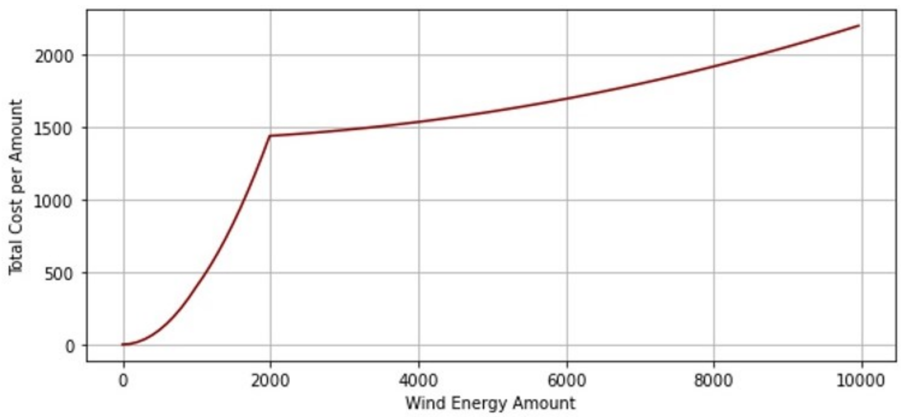

Figure 1.4: Wind cost function.

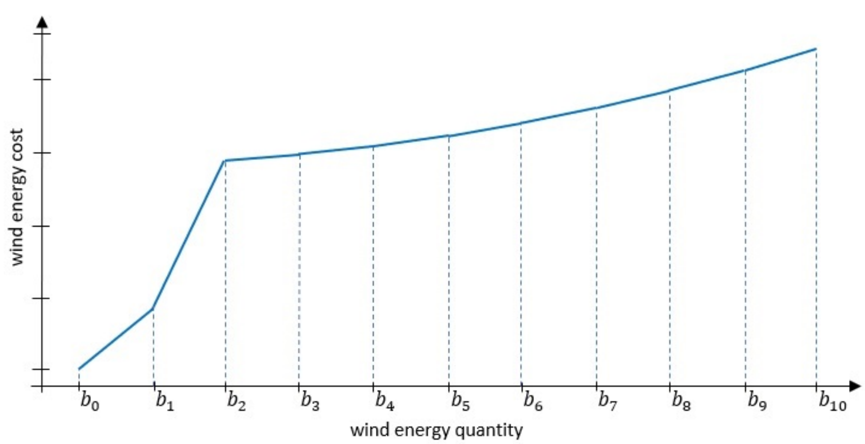

Figure 1.5: Piece-wise linearization of the wind cost function.

1.2.3 Nomenclature

Sets and indices

$\mathcal{G}$ Set of generators.

$\mathcal{T}$ Set of consecutive time periods $\{1, 2, ..., T\}$.

$\mathcal{S}$ Set of wind scenarios.

$\mathcal{B}$ Set of break points for discretizing the cost function.

Parameters

L_t Load (Demand) at time t (MW).

R_t^m Reserve at time t (MW) for model $m \in \{EVUC, SUC, PUC\}$.

$\overline{P}^g$ Maximum power output for generator g (MW).

$\underline{P}^g$ Minimum power output for generator g (MW).

RU^g Ramp-up rate for generator g (MW/h).

RD^g Ramp-down rate for generator g (MW/h).

SU^g Startup ramp rate for generator g (MW/h).

SD^g Shutdown ramp rate for generator g (MW/h).

UT^g Minimum up time for generator g (h).

DT^g Minimum down time for generator g (h).

C_{cs}^g Cold start cost of generator g.

C_{hs}^g Hot start cost of generator g.

C_{nl}^g No-load cost of generator g.

C_{ld} Lost demand cost.

${}^{b}C_p^g$ Production cost for each MW until break point b for generator g.

${}^{b}\overline{P}^g$ Maximum power output until break point b for generator g.

$\hat{p}_t$ Percentage of demand L_t that is available as wind power, we assume: $\hat{p}_1 = \hat{p}_2 = \ldots = \hat{p}_T$.

$\hat{p}_{ts}$ Percentage of demand L_t that is available as wind power in scenario s.

W_t Available wind capacity at time t.

Variables

u_t^g Binary, 1 if unit g is on at t; 0 otherwise

v_t^g Binary, 1 if unit g starts up at t; 0 otherwise

w_t^g Binary, 1 if unit g shuts down at t; 0 if otherwise

p_{ts}^g Power output of generator g at time t for scenario s.

$\bar{p}_{ts}^g$ Maximum available power output of generator g at time t for scenario s.

${}^{b}q_{ts}^g$ Power output of generator g at time t until break point b for scenario s.

c_t^g Binary, Cold start status of generator g at time t.

h_t^g Binary, Hot start status of generator g at time t.

$wind_{ts}$ Wind power used at time t for scenario s (MW).

S_{ts} Slack power at time t for scenario s (MW).

1.2.4 Expected Value Model

The expected value unit commitment (EVUC) model below minimizes the total operational cost z^{EVUC} considering wind generators in power supply. In this formulation, there are three binary vectors u^g, v^g, and w^g indicating commitment status, startup, and shutdown status of generator $g \in \mathcal{G}$. The vectors p^g, $\overline{p}^g$, and c^g denote the dispatch, maximum power output, and associated cost for generator g.

The total cost z^{UCEV} is composed from cold start C^g_{cs} and hot start cost C^g_{hs}, no load cost C^g_{nl} and the production cost ${}^bC^g_p$ for each generator $g \in G$ and time period $t \in \mathcal{T}$. The production cost is discretized by break points $b \in \mathcal{B}$ in order to capture the non-linearity by a piece-wise linear function. Constraints (1.8b) and (1.8c) ensure that produced energy satisfies hourly demand and reserve requirements for each time-period $t \in \mathcal{T}$. In constraints (1.8d) and (1.8e), the expected capacity of wind defines an upper bound on the usage of the wind power. The link between the commitment, startup, and shutdown status of generators is given in constraint (1.8f). The minimum up and down time of generators are given by constraints (1.8g) and (1.8h). Constraints (1.8i) and (1.8j) define the link between shutdown and hot/cold starts. Constraint (1.8k) makes sure that produced energy is within upper and lower capacity bounds of each generator. The effect of ramp-up and ramp-down rates on hourly electricity production is given in constrains (1.8l) and (1.8m). Constraints (1.8n) and (1.8o) include maximum power output for each break point b and the total power output at each time period t respectively.

$$z^{EVUC} = \min \sum_g \sum_t \Big(C^g_{cs} c^g_t + C^g_{hs} h^g_t + C^g_{nl} u^g_t + \sum_b {}^bC^g_p \, {}^bq^g_t \Big) \tag{1.8a}$$

$$\text{s.t.} \quad \sum_g p^g_t + wind_t \geq L_t \qquad \forall\, t \in \mathcal{T} \tag{1.8b}$$

$$\sum_g \bar{p}^g_t + wind_t \geq L_t + R^{EVUC}_t \qquad \forall\, t \in \mathcal{T} \tag{1.8c}$$

$$wind_t \leq \; E(W_t) \qquad \forall\, t \in \mathcal{T} \tag{1.8d}$$

$$E\left(W_t\right) = L_t \frac{\hat{p}_t}{2} \qquad \forall\, t \in \mathcal{T} \tag{1.8e}$$

$$u_t^g - u_{t-1}^g = v_t^g - w_t^g \qquad \forall\, t \in \mathcal{T} \tag{1.8f}$$

$$\sum_{i=t-UT+1}^{t} v_i^g \leq u_t^g \qquad \forall\, t \in [UT^g, T] \tag{1.8g}$$

$$\sum_{i=t-DT+1}^{t} w_i^g \leq 1 - u_t^g \qquad \forall\, t \in [DT^g, T] \tag{1.8h}$$

$$c_t^g \leq \sum_{i=DT}^{DT_C^g-1} w_t^g \qquad \forall\, t \in [DT^g, T] \tag{1.8i}$$

$$c_t^g + h_t^g = v_t^g \qquad \forall\, t \in [DT_C^g, T] \tag{1.8j}$$

$$\underline{P}^g u_t^g \leq p_t^g \leq \overline{p}_t^g \leq \overline{P}^g u_t^g \qquad \forall\, t \in \mathcal{T} \tag{1.8k}$$

$$\overline{p}_t^g - p_{t-1}^g \leq \; RU^g u_{t-1}^g + SU^g v_t^g \qquad \forall\, t \in \mathcal{T} \tag{1.8l}$$

$$\overline{p}_{t-1}^g - p_t^g \leq \; RD^g u_t^g + SD^g w_t^g \qquad \forall\, t \in \mathcal{T} \tag{1.8m}$$

$$0 \leq {}^b q_t^g \leq {}^b\overline{P}^g - {}^{b-1}\overline{P}^g \qquad \forall\, t \in \mathcal{T},\ \forall b \in \mathcal{B} \tag{1.8n}$$

$$p_t^g = \sum_b {}^b q_t^g \qquad \forall\, t \in \mathcal{T}. \tag{1.8o}$$

1.2.5 Stochastic Model

The stochastic unit commitment (SUC) model minimizes the total cost z^{SUC} taking into account the uncertainty of power output and power shortage by considering their associated average cost for each scenario $s \in \mathcal{S}$. This model formulates the standard two-stage unit commitment problem. The commitment decisions are made in the first stage so $u_s^g = u^g$ for all $s \in \mathcal{S}$ and $g \in \mathcal{G}$. Probability of scenario s is denoted by Pr_s. Constraints (1.9b) and (1.9c) include calculation of supply shortage in load and reserve requirement for each scenario. The usage of wind power is limited by the constraint (1.9d) where $\hat{p}_s$ is the percentage of demand that is predicted to be satisfied by wind power for each scenario $s \in \mathcal{S}$. Constraints (1.9e)-(1.9g) ensure that generators' upper/lower capacity limits, ramp-up/down rates, and startup/shutdown ramp rates are taken into account. Production cost curve break points are considered in constraints (1.9h) and (1.9i) for each scenario.

$$
\begin{aligned}
z^{SUC} = \min \sum_{g} \sum_{t} & (C^g_{cs} c^g_t + C^g_{hs} h^g_t + C^g_{nl} u^g_t) \\
& + \frac{1}{|S|} \sum_{s} \sum_{g} \sum_{t} \left(\sum_{b} {}^{b}C^g_p \, {}^{b}q^g_{ts} + C_{ld} S_{ts} \right)
\end{aligned} \tag{1.9a}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{g} p^g_{ts} + wind_{ts} + S_{ts} \geq L_t \; \forall \, t \in \mathcal{T}, && \forall \, s \in \mathcal{S} && (1.9b) \\
& \sum_{g} \overline{p}^g_{ts} + wind_{ts} + S_{ts} \geq L_t + R^{SUC}_t && \forall \, t \in \mathcal{T}, \; \forall \, s \in \mathcal{S} && (1.9c) \\
& wind_{ts} \leq \; L_t \hat{p}_{ts} && \forall \, t \in \mathcal{T}, \; \forall \, s \in \mathcal{S} && (1.9d) \\
& \underline{P}^g u^g_t \leq \; p^g_{ts} \leq \overline{p}^g_{ts} \leq \overline{P}^g u^g_t && \forall \, t \in \mathcal{T}, \; \forall \, s \in \mathcal{S} && (1.9e) \\
& \overline{p}^g_{ts} - p^g_{(t-1)s} \leq \; RU^g u^g_{t-1} + SU^g v^g_t && \forall \, t \in \mathcal{T}, \; \forall \, s \in \mathcal{S} && (1.9f) \\
& \overline{p}^g_{(t-1)s} - p^g_{ts} \leq \; RD^g u^g_t + SD^g w^g_t && \forall \, t \in [DT^g, T], \; \forall \, s \in \mathcal{S} && (1.9g) \\
& 0 \leq {}^{b}q^g_{ts} \leq {}^{b}\overline{P}^g - {}^{b-1}\overline{P}^g && \forall \, t \in \mathcal{T}, \; \forall b \in \mathcal{B}, \; \forall \, s \in \mathcal{S} && (1.9h) \\
& p^g_{ts} = \sum_{b} {}^{b}q^g_{ts} && \forall \, t \in \mathcal{T}, \; \forall \, s \in \mathcal{S} && (1.9i) \\
& \text{and } (1.8f), (1.8g), (1.8h), (1.8i), (1.8j) && \forall \, s \in \mathcal{S}. && (1.9j)
\end{aligned}
$$

1.2.6 Proposed Model

The objective function of the proposed model is the sum of wind cost $c(\cdot)$, lost demand cost, and all other costs z^{UCEV}. Constraints (1.10b) and (1.10c) are used to calculate shortage of supply for providing hourly demand and reserve requirements. Constraint (1.10d) prevents usage of wind power above the available capacity. Constraint (1.10e) is used to set the reserve to the maximum of demand and assigned wind.

$$
\begin{aligned}
z^{PUC} = \min \ & z^{EVUC} + c\,(wind_t) + C_{ld} S_t && && \text{(1.10a)}\\
\text{s.t.} \ & \sum_g p_t^g + wind_t \geq L_t && \forall\, t \in \mathcal{T} && \text{(1.10b)}\\
& \sum_g \bar{p}_t^g + wind_t + S_t \geq L_t + R_t^{PUC} && \forall\, t \in \mathcal{T} && \text{(1.10c)}\\
& wind_t \leq \ L_t \hat{p}_t && \forall\, t \in \mathcal{T} && \text{(1.10d)}\\
& R_t^{PUC} = \max\{L_t \hat{p}_t \ , \ wind_t\} && \forall\, t \in \mathcal{T} && \text{(1.10e)}\\
& \text{and } (1.8f - 1.8o). && && \text{(1.10f)}
\end{aligned}
$$

1.3 Computational Experiments

The performance of the proposed unit commitment (PUC) model (1.5) is assessed in terms of solution quality and CPU time with two alternative standard approaches: stochastic and expected value models that denoted by SUC and EVUC respectively. The SUC model minimizes the expected operational cost taking into account the uncertainty of power output and power shortage by considering their associated average cost for different scenarios. This model is a two-stage stochastic programming with unit commitment decisions in the first stage and real-time generation and load amount decisions in the second stage. The EVUC is a deterministic model that minimizes the total operational costs while limiting wind capacity in power supply by the expected value of wind profile over hourly time intervals in a given day. Hence, the wind capacity for all time periods is bounded by the fixed quantity. This simple though common in practice approach suffers from one major downfall, as it does not include the variability of the wind generation in each time period. The optimal generators' schedule is risky as it may utilize wind power to its full capacity while the wind realization might be lower. In the proposed model (PUC) the available wind is a fraction of hourly demand and it is available in full capacity, although the amount of its utilization is penalized by its uncertainty cost in the objective function. The novelty of this approach is recognition and quantification of the cost of uncertainty and varaibility of the wind power generation in every hour in the deterministic UC problem.

1.3.1 Computational Setup

All computational experiments were performed on a Lenovo T480s with Intel i7-8550U processor and 16 GB RAM running Windows 10. Gurobi 8.0.1 with Python interface is used for solving MIP and LP problems. Gurobi parameters were set to default for solving all models.

The data used in this study is the same UC test instances reported in Knueven et al., 2020b. This data-set was released by the US Federal Energy Regulatory Commission (FERC), which consist of two sets of generators: the Winter and Summer sets. The Summer data set is used in the computational study, that has 978 thermal units of which 504 have irredundant hourly ramping constraints. Thermal instances, real-time load, day-ahead reserves, and wind generation data for the year 2015 were utilized to formulate a 24 hour scheduling horizon unit commitment problem. A UC model with polynomial-sized convex hull formulation is used, which known as 3-bin model for a general thermal generator Knueven et al., 2018. In the data set there were no information about generators previous status. It is assumed that generators' current status were on and they were operating at minimum power. The set of instances are also available as part of the IEEE PES Power Grid Lib - Unit Commitment benchmark library (https://github.com/power-grid-lib/pglib-uc).

1.3.2 Assessment Procedure

In the assessment of the three models SUC, EVUC and PUC, it is assumed that the commitment decisions are made before the wind power generation is observed, and economic dispatch decisions are determined after the observation of wind outcomes.

The procedure started with building and solving the three unit commitment models to find the risk-aware day-ahead optimal commitment schedule of the generators for each model. A set of one hundred wind scenarios $\mathcal{S}$ is generated randomly, each one with different wind capacity ranging in the interval $[0, \overline{W}_t]$ for each hour. Then, while fixing generators' status to their optimal schedules, the associated dispatch problems is solved for each wind scenario $s \in \mathcal{S}$ to get their respective optimal cost and power output for both thermal and wind generators. The main purpose of solving the real-time dispatch problems in this work is to examine the efficiency of the proposed model in reducing the total operational costs. Random seed for the the scenario generation is fixed, and the same scenario set $\mathcal{S}$ is used to solve economic dispatch of each model. A different wind scenario set $\mathcal{S}_0$ is generated in order to build the SUC model to observed unpredictable nature of the wind generation output. Figure 1.6 illustrates the assessment procedure of the models.

1.3.3 Small Scale Results

In this section, the computational results on a small set of generators chosen from the FERC data-set is presented. The relative performance of the models in terms of run time and cost reduction is shown in Table 1.1. From the table it can be seen that the EVUC model is faster than the PUC due to the wind cost function calculation in the PUC. With the computational setup, the SUC is solvable by Gurobi within a reasonable amount of time (less than 1 hour) for sets of generators of maximum size 60. However, as expected, the SUC model outperforms the other two in terms of cost reduction with saving slightly more than 0.5 % on average with respect to EVUC model. The saving of PUC model relative to EVUC is approximately 0.5 % on average which is a significant achievement comparing with SUC. The proposed model, with significantly less computational costs, has a performance close to the stochastic model in terms of cost reduction. Figure 1.7 represents graphically the

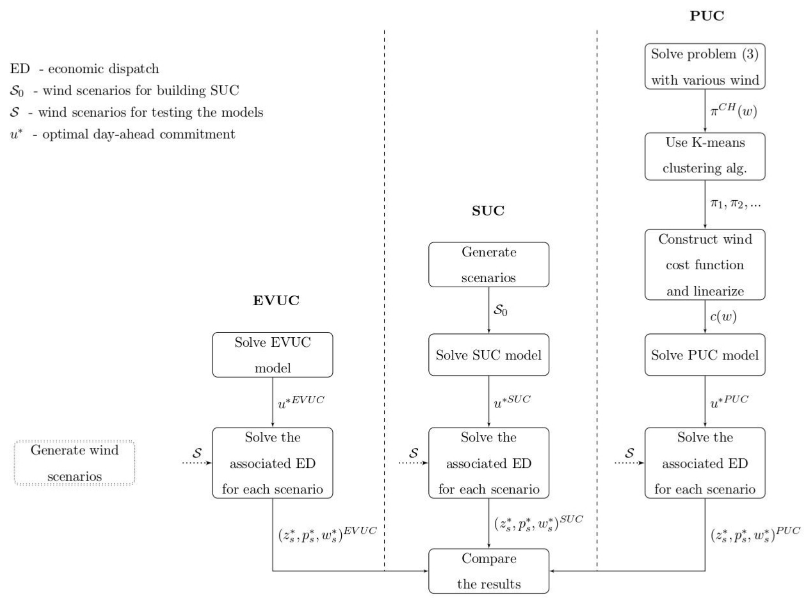

Figure 1.6: Solving procedure of EVUC, SUC and PUC models.

Table 1.1: Performance of the three models for a small set of generators

	Run times(sec)			Cost reduction relative to EVUC			
# of generators	EVUC	SUC	PUC	SUC($)	SUC(%)	PUC($)	PUC(%)
10	0	47	1	733	0.1	498	0.1
20	1	746	3	4795	0.4	4783	0.4
30	1	1060	4	9423	0.5	9302	0.5
40	2	1202	6	2219	0.4	2133	0.4
50	2	1723	9	9025	0.5	7747	0.5
60	2	1962	9	8819	0.6	8558	0.6
70	2	$>$ 10 h	11	-	-	21605	0.7
80	4	$>$ 10 h	30	-	-	25450	0.6
90	5	$>$ 10 h	41	-	-	58824	0.6

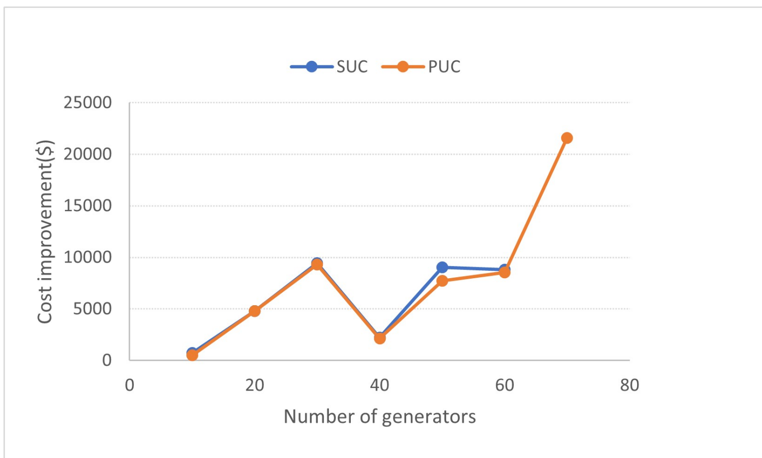

Figure 1.7: Cost improvement of SUC and PUC with respect to EVUC for small scale test set

relative performance of the SUC and PUC models with respect to the EVUC model in terms of cost saving at a small scale level.

1.3.4 Large Scale Results

In larger scale with about 1000 generators, only the performance of PUC and EVUC models could be compared since solving the SUC model is computationally intractable. To measure the performance of the PUC model, five levels of wind penetrations are analyzed with equal increments of 5% from 10% up to 30%. Table 1.2 shows the average across five separate instances for each level of wind penetration. The computational results which are reported in Tables 1.2, demonstrate that at each penetration level the PUC model outperforms the EVUC. The run times for building and linearizing the wind cost function is represented in WCF column. Note that computations for WCF are done sequentially, solving 240 linear programs varying the RHS of (1.3b) 10 times per time period. These computations can be easily parallelized, leading to significantly shorter wall-clock times. Surprisingly, the PUC instances actually solves faster than the EVUC model. With reasonable parallelization, it PUC with the WCF can take less wall-clock time than the EVUC model, leading to higher-quality schedules without impacting current operations.

1.4 Conclusion

In this study we proposed a risk-aware UC model in the presence of uncertain wind power generation. The price of wind variability is captured by constructing a wind cost function using CHP scheme. The proposed model mitigates the risk of wind integration in power supply, provides risk-aware optimal commitment schedule of generators, and minimizes the overall operational cost of the resource mix. The computational results demonstrated the effectiveness of the of the proposed model relative to the stochastic UC model and deterministic UC with wind uncertainty taken into account. An extension of the proposed model considers the temporal correlation of wind power production which deserves future research.

Table 1.2: Performance of EVUC and PUC models with various wind penetration level

	Average scenario costs ($)			Run times(sec)		
wind (%)	EVUC	PUC	Reduction	EVUC	PUC	WCF
10	73,406,115	73,073,499	332,616	160	120	325
15	71,589,676	71,431,922	157,754	180	141	376
20	69,960,659	69,910,216	50,443	197	130	433
25	68,663,582	68,565,969	97,613	209	123	502
30	67,436,116	67,232,866	203,250	222	119	587

Chapter 2

Symmetry Breaking Cut Generation for the Unit Commitment Problem

2.1 Nomenclature

2.1.1 Indices and Sets

$g \in \mathcal{G}$ Thermal generators.

$t \in \mathcal{T}$ Hourly Steps.

$[c, d) \in \mathcal{X}^g$ Feasible intervals of non-operation for generator g with respect to its minimum downtime, that is, $[c, d) \in \mathcal{T} \times \mathcal{T}$ such that $d \geq c + DT^g$.

2.1.2 Parameters

UT^g Minimum up time for generator g.

DT^g Minimum down time for generator g.

SU^g Startup ramp rate for generator g (MW/h).

SD^g Shutdown ramp rate for generator g (MW/h).

RU^g Ramp-up rate for generator g (MW/h).

RD^g Ramp-down rate for generator g (MW/h).

D_t Load (demand) at time t (MW).

R_t Spinning reserve at time t (MW).

$\overline{P}^g$ Maximum power output for generator g (MW).

$\underline{P}^g$ Minimum power output for generator g (MW).

TC^g Time down after which generator g goes cold, i.e., enters state S^g.

c^g Cost of production for generator g (\$/MWh).

$c^{R,g}$ Cost of generator g running and operating at minimum production $\underline{P}^g$ (\$/h).

$c^{C,g}$ Cold start cost of generator g (\$/h).

$c^{H,g}$ Hot start cost of generator g (\$/h).

2.1.3 Variables

u_t^g Binary, 1 if unit g is on at t; 0 otherwise.

v_t^g Binary, 1 if unit g starts up at t; 0 otherwise.

w_t^g Binary, 1 if unit g shuts down at t; 0 otherwise.

p_t^g Power above minimum for generator g at time t (MW).

$x_{[c,d)}^g$ Binary, 1 if generator g is uncommited between shutdown time at c and startup time at d, $[c, d) \in \mathcal{X}^g$; 0 otherwise.

2.2 Introduction

Unit commitment (UC) is an optimization problem, which is used to find optimal schedule and production level of thermal generating units over a given time period Bhardwaj et al., 2012. The committed units must satisfy technical constraints while producing sufficient energy to cover the forecasted load and reserve requirements. It has great practical importance because the effectiveness of generated schedules has a strong economical impact on power generation companies Viana and Pedroso, 2013.

UC is generally modelled as a mixed-integer linear problem (MILP) Knueven et al., 2017, and its has a strongly NP-hard complexity Bendotti et al., 2017. Some studies proposed heuristic methods to solve the problem López et al., 2013; Quan et al., 2014; Saber et al., 2006; Saksornchai et al., 2005; Simopoulos et al., 2006; Victoire and Jeyakumar, 2006; Yuan-Kang et al., 2013. But due to its economical significance sacrificing from optimality is not preferred. Therefore, many researches focused on tightening MILP formulation to have a better LP relaxation of the problem. This approach was proven to be computationally effective and studied by Gentile et al., 2017; Knueven et al., 2018; Morales-España et al., 2015; Morales-España et al., 2013; Ostrowski et al., 2012.

Similar to other MILP problems, UC problem gets more computationally difficult if it involves symmetrical property. In this context, symmetry means having at least two identical generators with same performance settings and cost parameters. A single location may have multiple generators of the same type which will lead to symmetry in the MILP formulation. In this case, the solution process could take extra time as the solver may explore different commitment combinations for identical generators. From the cost reduction perspective, however, this search process is redundant because symmetry in the problem can cause having multiple optimal solutions.

In the UC literature, several methods presented to handle the symmetry. One approach is to use an alternative branching strategies while examining branch-and-bound tree. If the problem's symmetry group have special structure orbital branching could be strengthened Ostrowski et al., 2015; Ostrowski et al., 2011. Such methods, however, require customized branching and are difficult to implement using commercial solvers. Another method used in literature is to use symmetry breaking cuts to eliminate symmetry in UC problem, see Alemany et al., 2014; Alemany et al., 2016; Fu et al., 2019; Lima and Novais, 2016. In these studies, the commitment status of identical generators were put in hierarchical order to prioritize their utilization. The major drawback of this approach is to have extensive number of hierarchical constraints which would increase the size of the LP relaxation. It was reported that for large scale UC instances performance of solver decreases Lima and Novais, 2016.

For solving UC problem with symmetrical property, an alternative approach in the literature is aggregating identical thermal generators into a single unit. It consist three simple section: aggregation, solving the aggregated model, and disaggregation to create schedules for individual units Sen and Kothari, 2002. Several studies (Garcia-Gonzalez et al., 2008; Palmintier and Webster, 2011; Shortt and O'Malley, 2010) used aggregation to speed up the solution process, but they haven't considered all technical constraints of the generators, therefore, the found schedules were sub-optimal. An exact aggregation was done by Knueven et al., 2017, they showed the conditions under which such an aggregation can be done exactly. Their approach was only effective for fast-ramping thermal generators, where the ramping constraints are redundant. Decomposable formulations for slow-ramping generators tend to be very large, such that the benefits from the aggregation outweigh the cost of the additional variables and constraints. Slow-ramping generators are are generators thac cannot ramp to full power in one time interval. Historically this has not been a problem as slow-ramping generators, such as coal burning units, make up a small share of the production in many markets (such as CAISO and MISO). However, moving to sub-hourly time horizons, such as 15 minute UC, would drastically increase the number of slow-ramping generators, necessitating the need for new methods.

In this paper, we propose an aggregate and cut method that guaranties feasibility. For the cases which a feasible disaggregation is impossible we a feasibility cut to prevent such solutions. To improve the solution time further, we identify a pattern of commonly used feasibility cuts that can be added to the aggregation model prior to the cut-generation process.

2.3 Symmetry in the Unit Commitment Problem and its Effect

2.3.1 Base "3-Bin" Model

UC problem has multiple formulations presented in the literature Knueven et al., 2020b. A "3-bin" UC MILP formulation are typically expressed in this form Carrión and Arroyo, 2006a; Ostrowski et al., 2012:

$$\min \sum_{g\in\mathcal{G}}\sum_{t\in\mathcal{T}} c^g(u_t^g, p_t^g) \tag{2.1a}$$

$$\text{s.t.} \sum_{g\in\mathcal{G}} p_t^g = D_t \qquad \forall t\in\mathcal{T}, \forall g\in\mathcal{G} \tag{2.1b}$$

$$\underline{P}^g u_t^g \leq p_t^g \leq \overline{P}^g u_t^g \qquad \forall t\in\mathcal{T}, \forall g\in\mathcal{G} \tag{2.1c}$$

$$p_t^g - p_{t-1}^g \leq RU^g u_{t-1}^g + SU^g v_t - (\underline{P}^g - RU^g) w_t \qquad \forall t\in\mathcal{T}, \forall g\in\mathcal{G} \tag{2.1d}$$

$$p_{t-1}^g - p_t^g \leq RD^g u_t^g + SD^g w_t - (\underline{P}^g - RD^g) v_t \qquad \forall t\in\mathcal{T}, \forall g\in\mathcal{G} \tag{2.1e}$$

$$u_t^g - u_{t-1}^g = v_t^g - w_t^g \qquad \forall t\in\mathcal{T}, \forall g\in\mathcal{G} \tag{2.1f}$$

$$\sum_{i=t-UT^g+1}^{t} v_i^g \leq u_t^g \qquad \forall t\in[UT^g, T], \forall g\in\mathcal{G} \tag{2.1g}$$

$$\sum_{i=t-DT^g+1}^{t} w_i^g \leq 1-u_t^g \qquad \forall t\in[DT^g,T], \forall g\in\mathcal{G} \qquad (2.1h)$$

$$u_t^g, v_t^g, w_t^g \in\{0,1\}, p_t^g\in\mathbb{R}_+ \qquad \forall t\in\mathcal{T}, \forall g\in\mathcal{G} \qquad (2.1i)$$

where (2.1a) shows sum of the costs associated with generator g producing an output p_t^g over the scheduling horizon. Constraints (2.1b) makes sure that all the demands are satisfied, while generated power doesn't exceed from the its capacity limitations (2.1c). Constraints (2.1d) and (2.1e) enforce ramping limits. Constraints (2.1f) enforce the logical connection between the variables u, v, and w, and Constraints (2.1g) and (2.1h) enforce minimum up and down times.

A symmetry in the UC may happen due to utilization of identical generators. The two generator are called to be identical if all their specifications are same. These specification are ramp-up/ramp-down rates, system-up/system-down rates, minimum/maximum production levels, production cost, and so on. Note that identical generators can have different statusus at the start of the planning horizon. Having identical generators in the UC problems solved by ISO is common, because a single power generator company may have two or more generator of same type in their power plant.

Consider a small example with three group of identical generators with eight, ten, and twelve generators in each group. If the optimal solution requires half of the generators from each group to be committed, then number of combinations these generators to be chosen will be:

$$\binom{8}{4}\binom{10}{5}\binom{12}{6}$$

In this example number of alternative solutions will be more than sixteen million. It is obvious that there is no benefit of checking every alternative solution. However, without any symmetry breaking cuts the solver may enumerate all these cases and take longer time to solve it, and in some cases it won't be able to solve within hours.

There is a need for a new approach which exploits symmetry and prevent redundant calculations. By avoiding symmetries, we can reduce the search space significantly, therefore, improve the computation time.

2.3.2 Symmetry Exploiting Relaxation

For solving UC problem with identical generators an aggregated model could be used Knueven et al., 2017. In the aggregated model the binary decision variables for each identical generator group (u, v, w) are set to be integer and the generator parameters $(\underline{P}, \overline{P}, SU, SD)$ are adjusted accordingly. If the generators are fast-ramping, in other words, they have redundant ramping constraint, then the traditional 3-bin formulation could be used, when $UT \geq 2$ Knueven et al., 2017:

$$\underline{P}u_t \leq p_t \qquad \forall t \in \mathcal{T} \tag{2.2a}$$

$$p_t \leq \overline{P}u_t + (SU - \underline{P})v_t + (SD - \overline{P})w_{t+1} \qquad \forall t \in \mathcal{T} \tag{2.2b}$$

$$u_t - u_{t-1} = v_t - w_t \qquad \forall t \in \mathcal{T} \tag{2.2c}$$

$$\sum_{i=t-UT+1}^{t} v_i \leq u_t \qquad \forall t \in [UT, T] \tag{2.2d}$$

$$\sum_{i=t-DT+1}^{t} w_i \leq 1 - u_t \qquad \forall t \in [DT, T] \tag{2.2e}$$

$$u_t, v_t, w_t \in \{0, 1\},\ p_t \in \mathbb{R}^+ \qquad \forall t \in \mathcal{T} \tag{2.2f}$$

Here, the Constraints (2.2c), (2.2d), (2.2e) are totally unimodular Malkin and Wolsey, 2003, and they have integer decomposition property Baum and Trotter, 1978. Knueven and Ostrowski, (2018) showed that it is be possible to generate feasible schedule by disaggregating the found solution in the aggregation model.

Suppose that, we partition the set of generators $\mathcal{G}$ into three subset: non-identical generators $(\mathcal{G}_{\nsim})$, identical with slow-ramping generators $(\mathcal{G}^S)$, and identical with fast-ramping generators $(\mathcal{G}^F)$. Here, $\mathcal{G}_{\nsim} = \{g \in \mathcal{G} \mid g \nsim g' \ \forall g' \in \mathcal{G} \backslash \{g\}\}$, $\mathcal{G}_{\sim} = \mathcal{G} \backslash \mathcal{G}_{\nsim}$,

$\mathcal{G}^F = \{g \in \mathcal{G}_\sim \,|\, RU^g, RD^g \geq (\overline{P}^g - \underline{P}^g)\}$, and $\mathcal{G}^S = \mathcal{G}_\sim \backslash \mathcal{G}^F$. After adjusting generator parameters properly for the generator cluster ($\overline{\mathbf{P}}^\mathcal{K}, \underline{\mathbf{P}}^\mathcal{K}, \mathbf{SU}^\mathcal{K}, \mathbf{SD}^\mathcal{K}, \mathbf{RU}^\mathcal{K}, \mathbf{RD}^\mathcal{K}$), the aggregated model for slow-ramping generator could be formulated as following master problem:

$$MP \quad \sum_{\mathcal{K}\in\mathcal{G}^S}\sum_{t\in\mathcal{T}}\Bigg(c^\mathcal{K}P_t^\mathcal{K}+c^{R,\mathcal{K}}U_t^\mathcal{K}-c^{C,\mathcal{K}}V_t^\mathcal{K} + (c^{H,\mathcal{K}}-c^{C,\mathcal{K}})\Bigg(\sum_{t'=t-T^{s,\mathcal{K}}+1} X_{[t',t)}^\mathcal{K}\Bigg)\Bigg) \tag{2.3a}$$

subject to

$$\sum_{\mathcal{K}\in\mathcal{G}^S}(P_t^\mathcal{K}+\underline{\mathbf{P}}^\mathcal{K}U_t^\mathcal{K})=D_t \qquad \forall t\in\mathcal{T} \tag{2.3b}$$

$$\sum_{\mathcal{K}\in\mathcal{G}^S} r_t^g \geq R_t \qquad \forall t\in\mathcal{T} \tag{2.3c}$$

$$P_t^\mathcal{K}+R_t^\mathcal{K}\leq(\overline{\mathbf{P}}^\mathcal{K}-\underline{\mathbf{P}}^\mathcal{K})U_t^\mathcal{K} - (\overline{\mathbf{P}}^\mathcal{K}-\mathbf{SU}^\mathcal{K})V_t^\mathcal{K} \qquad \forall t\in\mathcal{T}, \forall\mathcal{K}\in\mathcal{G}_1^S \tag{2.3d}$$

$$P_t^\mathcal{K}+R_t^\mathcal{K}\leq(\overline{\mathbf{P}}^\mathcal{K}-\underline{\mathbf{P}}^\mathcal{K})U_t^\mathcal{K} - (\overline{\mathbf{P}}^\mathcal{K}-\mathbf{SD}^\mathcal{K})W_{t+1}^\mathcal{K} \qquad \forall t\in\mathcal{T}, \forall\mathcal{K}\in\mathcal{G}_1^S \tag{2.3e}$$

$$P_t^\mathcal{K}+R_t^\mathcal{K}\leq(\overline{\mathbf{P}}^\mathcal{K}-\underline{\mathbf{P}}^\mathcal{K})U_t^\mathcal{K} - (\overline{\mathbf{P}}^\mathcal{K}-\mathbf{SU}^\mathcal{K})V_t^\mathcal{K} - (\overline{\mathbf{P}}^\mathcal{K}-\mathbf{SD}^\mathcal{K})W_{t+1}^\mathcal{K} \qquad \forall t\in\mathcal{T}, \forall\mathcal{K}\in\mathcal{G}_{>1}^S \tag{2.3f}$$

$$P_t^\mathcal{K}+R_t^\mathcal{K}-P_{t-1}^\mathcal{K}\leq\mathbf{RU}^\mathcal{K} \qquad \forall t\in\mathcal{T}, \forall g\in\mathcal{G}^S \tag{2.3g}$$

$$P_{t-1}^\mathcal{K}-P_t^\mathcal{K}\leq\mathbf{RD}^\mathcal{K} \qquad \forall t\in\mathcal{T}, \forall g\in\mathcal{G}^S \tag{2.3h}$$

$$U_t^\mathcal{K}-U_{t-1}^\mathcal{K}=V_t^\mathcal{K}-W_{t+1}^\mathcal{K} \qquad \forall t\in\mathcal{T}, \forall\mathcal{K}\in\mathcal{G}^S \tag{2.3i}$$

$$\sum_{i=t-UT^\mathcal{K}+1}^{t} V_i^\mathcal{K}\leq U_t^\mathcal{K} \qquad \forall t\in[UT^\mathcal{K},T], \forall\mathcal{K}\in\mathcal{G}^S \tag{2.3j}$$

$$\sum_{i=t-DT^{\mathcal{K}}+1}^{t} W_i^{\mathcal{K}} \leq |\mathcal{K}| - U_t^{\mathcal{K}} \qquad \forall t \in [DT^{\mathcal{K}}, T], \forall \mathcal{K} \in \mathcal{G}^S \tag{2.3k}$$

$$\sum_{t'=t-TC^{\mathcal{K}}+1}^{t-DT^{\mathcal{K}}} X_{[t',t)}^{\mathcal{K}} \leq V_t^{\mathcal{K}} \qquad \forall t \in \mathcal{T}, \forall \mathcal{K} \in \mathcal{G}^S \tag{2.3l}$$

$$\sum_{t'=t+DT^{\mathcal{K}}}^{t+TC^{\mathcal{K}}-11} X_{[t',t)}^{\mathcal{K}} \leq W_t^{\mathcal{K}} \qquad \forall t \in \mathcal{T}, \forall \mathcal{K} \in \mathcal{G}^S \tag{2.3m}$$

$$P_t^{\mathcal{K}}, R_t^{\mathcal{K}} \in \mathbb{R}^+ \qquad \forall t \in \mathcal{T}, \forall \mathcal{K} \in \mathcal{G}^S \tag{2.3n}$$

$$U_t^{\mathcal{K}}, V_t^{\mathcal{K}}, W_t^{\mathcal{K}} \in \{0, \ldots, |\mathcal{K}|\} \qquad \forall t \in \mathcal{T}, \forall \mathcal{K} \in \mathcal{G}^S \tag{2.3o}$$

$$X_{[t',t)}^{\mathcal{K}} \in \{0, \ldots, |\mathcal{K}|\} \qquad \forall [t',t) \in \mathcal{X}^{\mathcal{K}}, \forall \mathcal{K} \in \mathcal{G}^S \tag{2.3p}$$

Here, the objective function (2.3a) minimizes total cost production, while Constraints (2.3b) and (2.3c) ensure demand and reserve requirements are met, and the remaining constraints describe the technical constraints for the aggregated generators.

However, as mentioned above, this method works with generator without any ramping constraint. If otherwise, the aggregated solution may not be disaggreageted into a feasible schedule. Consider the following example with two generators identical parameters. Assume that, $RU = RD = 10$, $SU = SD = \underline{P} = 20$, and $\overline{P} = 40$. In aggregated model we will have the following transformation: $U^{\mathcal{K}}, V^{\mathcal{K}}, W^{\mathcal{K}} \in \{0, 1, 2\}$, where $RU^{\mathcal{K}} = RD^{\mathcal{K}} = 20$, $SU^{\mathcal{K}} = SD^{\mathcal{K}} = \underline{P}^{\mathcal{K}} = 40$, and $\overline{P}^{\mathcal{K}} = 80$. Given these settings the following solution is feasible for the aggregated UC model:

$$u^{\mathcal{K}} = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}, \quad p^{\mathcal{K}} = \begin{pmatrix} 80 \\ 60 \\ 40 \end{pmatrix}. \tag{2.4}$$

At time period 1 both generators must work at their maximum capacity, and at period 1 one generator must work at its maximum capacity. Then, there is at least one generator that works at its full capacity in these 3 periods. By decomposing this solution, we get the following:

$$u = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad p = \begin{pmatrix} 40 & 40 \\ 40 & 20 \\ 40 & 0 \end{pmatrix}. \tag{2.5}$$

Clearly, this disaggregated solution is not feasible, because the second generator violates the ramping constraint.

2.4 Cut Generation Approach

2.4.1 General Cut Approach

To model the UC with non-redundant ramping constraints packing dispatch polytopes could be used to reformulate the constraints in a compact (polynomial-sized) formulation, see Knueven et al., 2018. If a binary decision variable, $y_{[a,b]}$ shows generator is committed in time interval $[a, b]$, and $\mathcal{I}$ shows all possible intervals, then for every $t \in \mathcal{T}$ constraints can be reformulated as:

$$A^{[a,b]}p^{[a,b]} + \overline{A}^{[a,b]}\overline{p}^{[a,b]} \leq y_{[a,b]}\overline{b}^{[a,b]} \qquad \forall [a,b] \in \mathcal{I} \tag{2.6a}$$

$$\sum_{[a,b]\in\mathcal{I}} p^{[a,b]} = p \tag{2.6b}$$

$$\sum_{[a,b]\in\mathcal{I}} \overline{p}^{[a,b]} = \overline{p} \tag{2.6c}$$

$$\sum_{\{[a,b]\in\mathcal{I} \,|\, t\in[a,b+DT]\}} y_{[a,b]} \leq 1 \tag{2.6d}$$

$$y_{[a,b]} \geq 0 \qquad \forall [a,b] \in \mathcal{I} \tag{2.6e}$$

$$(p^{[a,b]}, \overline{p}^{[a,b]}) \in \mathbb{R}^{2T}_{+} \qquad \forall [a,b] \in \mathcal{I} \tag{2.6f}$$

Here, $p_t^{[a,b]}$ represents power produced by the generator at time t, $\overline{p}_t^{[a,b]}$ represents the maximum available power at time t. This reformulation could be used to used to add a feasibility cut to the aggregated model.

Please note that, due to involvement of the large number of possible intervals ($\mathcal{I}$), formulation (2.6) is too big and computationally hard to solve. Therefore, instead of using it directly we use it a tool for generating cuts.

Let $(p^*, \overline{p}^*, U^*, V^*, W^*)$ be a solution for the aggregated three-bin model. Let e be the properly sized vector of 1's. Consider the following sub problem.

$$SP \qquad z^* = min(z) \tag{2.7a}$$

subject to

$$\alpha \qquad A_t^{[a,b]} p_t^{[a,b]} + \overline{A}_t^{[a,b]} \overline{p}_t^{[a,b]} \leq \leq y_{[a,b]} \overline{b}^{[a.b]} + ze \qquad \forall [a,b] \in \mathcal{I}, \forall t \in \mathcal{T} \tag{2.7b}$$

$$\beta \qquad \sum_{\{[a,b] \in \mathcal{I} | t \in [a, b+DT]\}} y_{[a,b]} \leq |\mathcal{K}| + z \qquad \forall t \in \mathcal{T} \tag{2.7c}$$

$$\gamma \qquad \sum_{[a,b] \in \mathcal{I}} p_t^{[a,b]} = p_t^* \qquad \forall t \in \mathcal{T} \tag{2.7d}$$

$$\delta \qquad \sum_{[a,b] \in \mathcal{I}} \overline{p}_t^{[a,b]} = \overline{p}_t^* \qquad \forall t \in \mathcal{T} \tag{2.7e}$$

$$\epsilon \qquad \sum_{\{[a,b] \in \mathcal{I} | t \in [a,b]\}} y_{[a,b]} = U_t^* \qquad \forall t \in \mathcal{T} \tag{2.7f}$$

$$\zeta \qquad \sum_{\{[a,b] \in \mathcal{I} | t = a\}} y_{[a,b]} = V_t^* \qquad \forall t \in \mathcal{T} \tag{2.7g}$$

$$\eta \qquad \sum_{\{[a,b] \in \mathcal{I} | t = b+1\}} y_{[a,b]} = W_t^* \qquad \forall t \in \mathcal{T} \tag{2.7h}$$

$$z \in \mathbb{R}_+, (p^{[a,b]}, \overline{p}^{[a,b]}) \in \mathbb{R}_+^{2T}, y_{[a,b]} \in \mathbb{R}_+ \qquad \forall [a,b] \in \mathcal{I} \tag{2.7i}$$

If the solution of the aggregated three-bin model $(p^*, \overline{p}^*, U^*, V^*, W^*)$ is feasible, in other words decomposible, then the optimal solution for model (2.7) would be zero. Otherwise, if $z^* > 0$ then we can use the duals of constraints $(\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta)$ to generate a cut for the aggregated three-bin model (Master problem). By strong duality, $z^* = (\beta^*)^T e + (\gamma^*)^T p^* + (\delta^*)^T \overline{p}^* + (\epsilon^*)^T U^* + (\zeta^*)^T V^* + (\eta^*)^T W^* > 0$. Therefore, by adding cut

$$(\beta^*)^T e + (\gamma^*)^T p^* + (\delta^*)^T \overline{p}^* + (\epsilon^*)^T U^* + (\zeta^*)^T V^* + (\eta^*)^T W^* \leq 0,$$

we can cut of the solution $(p^*, \overline{p}^*, U^*, V^*, W^*)$ which is not a feasible solution for a decomposed model. We will refer this cut as a feasibility cut (FC). These feasibility cuts will be added to the master problem iteratively until z^* for each group of identical generators are zero. The cut generation procedure is visualized in Figure 2.1.

2.4.2 Enhancements

Commonly Found Cuts

In order to reduce the computational time, the generated feasibility cuts can be analyzed to detect their pattern. Then, they could be added to the master problem to decrease the number of the solved sub problems. Indeed, it was observed that some of the feasibility cuts follow a specific pattern. Assume that, A and B are defined as following:

$$A = \sum_{i=0}^{min(m,t)+1} (SU^{\mathcal{K}} - \underline{P}^{\mathcal{K}} + iRU^{\mathcal{K}} - (\overline{P}^{\mathcal{K}} - \underline{P}^{g})) v_{t-i}^{\mathcal{K}}$$

$$B = \sum_{j=0}^{min(n,T-t+1)+1} (SU^{\mathcal{K}} - \underline{P}^{g} + jRD^{\mathcal{K}} - (\overline{P}^{\mathcal{K}} - \underline{P}^{\mathcal{K}})) w_{t+1+j}^{\mathcal{K}}$$

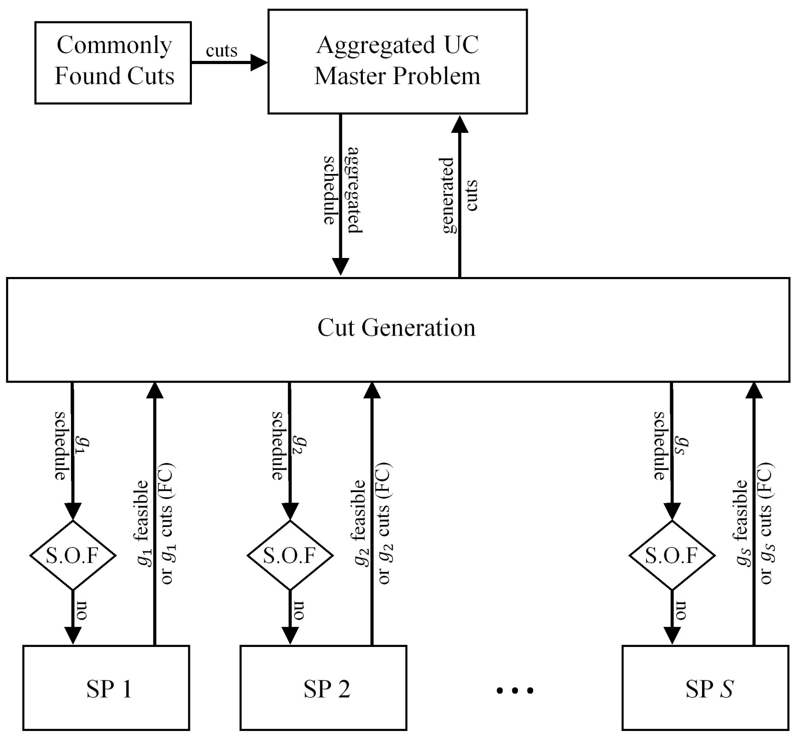

Figure 2.1: Visualization of enhanced cut generation procedure.

where

$$m = min(\left\lceil \frac{\overline{P}^g - SU^{\mathcal{K}}}{RU^{\mathcal{K}}} \right\rceil, UT^{\mathcal{K}} + DT^{\mathcal{K}} + 1)$$
$$n = min(\left\lceil \frac{\overline{P}^{\mathcal{K}} - SD^{\mathcal{K}}}{RD^{\mathcal{K}}} \right\rceil, UT^{\mathcal{K}} + DT^{\mathcal{K}})$$

Then, the pattern of the cuts could be shown by these two group of constraints:

$$p_t^{\mathcal{K}} + r_t^{\mathcal{K}} \leq (\overline{P}^{\mathcal{K}} - \underline{P}^{\mathcal{K}})u_t^{\mathcal{K}} + A \qquad \forall \mathcal{K} \in \mathcal{G}^S, \forall t \in \mathcal{T} \qquad (2.9a)$$
$$p_t^{\mathcal{K}} + r_t^{\mathcal{K}} \leq (\overline{P}^{\mathcal{K}} - \underline{P}^{\mathcal{K}})u_t^{\mathcal{K}} + B \qquad \forall \mathcal{K} \in \mathcal{G}^S, \forall t \in \mathcal{T} \qquad (2.9b)$$

In the first cut, the generators' ramp-up limit (RU), startup rate (SU), and total operation time were used to set the upper bound to the maximum possible energy output, whereas, ramp-down limit (RD), shutdown rate (SD) were used for the second. Here, $\mathcal{G}^S$ represents the group of slow-ramping generators which have identical characteristics.

Simultaneous On/Off

The computational time could be further decreased by considering a special case with on/off status of the generators. We first provide a theorem regarding the relationship between aggregated schedules and the power output of identical generators.

Theorem 2.1. *Consider identical generators $g_1, g_2, \ldots, g_n \in \mathcal{G}^S$, and assume that in optimal solution of the aggregated model (MP) all of them are on or off at the same time. Then, the solution to the aggregated model is feasible in the disaggregated model.*

Proof. Assume that there are n identical generators with the set $\mathcal{N}$ and $p_t^{\mathcal{K}}$ shows the aggregated power generation at time t, where $p_t^{\mathcal{K}} = \sum_{g \in \mathcal{N}} p_t^g$. Similarly, $RU^{\mathcal{K}}$ and

$RD^{\mathcal{K}}$ shows the aggregated ramping limits, where $RU^{\mathcal{K}} = n \times RU$ and $RD^{\mathcal{K}} = n \times RD$. After solving the master problem (MP), the resulting aggregated generator schedule should satisfies the following ramping constraints.

$$p_t^{\mathcal{K}} - p_{t-1}^{\mathcal{K}} \leq RU^{\mathcal{K}} \qquad \forall t \in \mathcal{T} \tag{2.10a}$$

$$p_{t-1}^{\mathcal{K}} - p_t^{\mathcal{K}} \leq RD^{\mathcal{K}} \qquad \forall t \in \mathcal{T} \tag{2.10b}$$

If all generators are on and off at the same time, we can choose the production of each individual generator in a way that for every time period t the aggregated production $p_t^{\mathcal{K}}$ is equally divided between n identical generators, where $p_t^g = p_t^{\mathcal{K}}/n$. In this case, by dividing both side of inequalities in (2.10) we will get:

$$p_t^g - p_{t-1}^g \leq RU \qquad \forall g \in \mathcal{N}, \forall t \in \mathcal{T} \tag{2.11a}$$

$$p_{t-1}^g - p_t^g \leq RD \qquad \forall g \in \mathcal{N}, \forall t \in \mathcal{T} \tag{2.11b}$$

Inequalities in (2.11) show that the disaggregated generator schedule is also satisfying ramping limitations. ■

From the Theorem 2.1, if we have aggregated schedule with all generators are on or off at the same time (S.O.F), then there is no need to solve the sup problem (SP) in this case, as it already leads to a feasible disaggregated schedule. For improving the computational time, the solution approach could be updated to avoid solving the sub problem if such special case is detected.

2.5 Performance Comparison

2.5.1 Comparison Overview

We demonstrate the performance of the new reformulation which uses aggregation with the "3-bin" formulation proposed by Knueven et al., 2020a. Note the proposed reformulation will have three main section. In first section, we have constraints for non-identical thermal generators ($\mathcal{G}_{\not\sim}$), in second section we will have the proposed approach for identical slow-ramping generators ($\mathcal{G}^S$), and in third section we will use the approach proposed by Knueven et al., 2017 to model fast-ramping generators ($\mathcal{G}^F$). These three models have common demand and reserve constraints, whereas their corresponding production costs are minimized in the objective function (see the Appendix for details). From now on, we will be referring this reformulated model as “R”. We will refer the improved model as “R+SOF+CFC”. In the improved model we are considering simultaneous on/off cases (SOF) and adding commonly found cuts (CFC) to the model for increasing the performance.

2.5.2 Data Sets Description

Two unit commitment test sets from the literature were used to test the effectiveness of the proposed aggregation approach. The first data set was taken from Ostrowski et al., 2012. We refer to these instances as the ”Academic” instances, they are constructed by replicating the thermal generators originally introduced in Carrión and Arroyo, 2006b and Kazarlis et al., 1996. Replication of these generator have been used to generate large UC instances in many UC studies Borghetti et al., 2001; Borghetti et al., 2003; Feizollahi et al., 2015; Frangioni and Gentile, 2006, 2009; Frangioni et al., 2008a, 2008b; Jabr, 2012; Morales-España et al., 2015; Yang et al., 2015; Yang et al., 2014. There are 8 different generators in these data set with three fast-ramping generators. We create 20 instance where total number of generators in each instance range from 28 to 187 by replicating the generators, see Table 2.1. The reserve level was set to be 3% of system demand, where hourly demand was set to be a percentage of total system capacity ranging from 58% to 91%. Scheduling horizon of 24 hour was used with time unit set to 1 hour. For further detail please refer to Knueven et al., 2017.

The second set UC instances are based on real-world data acquired from the California Independent System Operator (CAISO). There 20 instances with 610 thermal generators each. Among them 400 are unique generators, 210 are fast-ramping generators, and there are no identical slow-ramping generators in the hourly UC instances. However, when considering a 15-minute time horizon, many of these generators became slow-ramping. . Generators parameters (RU, RD, UT, DT) were modified in accordance with the unit time, which eliminated all fast-ramping generators. The resulting data set had 210 slow-ramping generator which can be clustered into 66 groups. Eliminating fast-ramping generators from the data set helped to test the proposed approach solely without taking benefit of computation performance improvement introduced by Knueven et al., 2017. Spinning reserve was set to 3% of demand. Wind supply with 0%, 1%, 3%, and 5% of the demand were used, where they at as negative demand. Besides, a theoretical 40% wind supply was tested to check its effect on solution time. For further detail about these data set please refer to Knueven et al., 2020a.

All computational experiments were conducted on a Lenovo T480s laptop with Intel i7-8550U processor, for a total of 4 cores, 8 threads, an 16GB of RAM, running on the Windows 10 operating system. The Gurobi 8.0.1 MILP solver was used in all experiments.

2.5.3 Academic Instances

For all test cases a time limit of two hours (7200 sec.) was imposed. If the solution couldn't be found in given time we report the percentage optimality gap in the parenthesis. The Academic instances are computationally easy compared to CAISO instances. Except three instances the all instances in the data set are solvable by "3-bin" model. Average run time of "3-bin" model for the seven solvable instances are approximately 381 seconds, see Table 2.2.

Table 2.1: Academic Instances: Number of Generators of Each Type

Problem	Generator								Total Gens
	1	2	3	4	5	6	7	8	
1	12	11	0	0	1	4	0	0	28
2	13	15	2	0	4	0	0	1	35
3	15	13	2	6	3	1	1	3	44
4	15	11	0	1	4	5	6	3	45
5	15	13	3	7	5	3	2	1	49
6	10	10	2	5	7	5	6	5	50
7	17	16	1	3	1	7	2	4	51
8	17	10	6	5	2	1	3	7	51
9	12	17	4	7	5	2	0	5	52
10	13	12	5	7	2	5	4	6	54
11	46	45	8	0	5	0	12	16	132
12	40	54	14	8	3	15	9	13	156
13	50	41	19	11	4	4	12	15	156
14	51	58	17	19	16	1	2	1	165
15	43	46	17	15	13	15	6	12	167
16	50	59	8	15	1	18	4	17	172
17	53	50	17	15	16	5	14	12	182
18	45	57	19	7	19	19	5	11	182
19	58	50	15	7	16	18	7	12	183
20	55	48	18	5	18	17	15	11	187

On the other hand, reformulated model was able to solve all test instances with the mean value of 28.4 seconds. it appears that, the solved sub-problem (SP) and added feasibility cuts (FC) are not related to total number of generators in each instances. The improved model was able to solve all instances even faster with an average of 16.7 seconds. The added commonly found cuts (CFC) helped to reduce number of sup-problems and added feasibility cuts. In all instances the reformulation helped to reduce the scale of the problem significantly. We reported number of the rows (Row), columns (Col), and non zero entries (NZ) in the pre-processed model in Table 2.3. For the "3-bin" model we can see that the scale in increasing as the number of identical generators increase in the instances. However, it had slow to no effect to the reformulated model. The improved model had slightly larger scale which is correlated with added CFC.

2.5.4 CAISO Instances

For CAISO instances only 6 of the 20 instances were able to be solved using the base formulation, see Table 2.4. However, all of the instances could be solved by the proposed model with an average time of 1435 seconds. In each of the instances, the number of the added feasibility cuts is no more than 10. One reason for having less feasibility cuts is added commonly found cuts, as CFC includes some of the feasibly cuts. The second reason is that the majority of the solutions found by sub problems were decomposable; approximately half of those cases could be detected and avoided by checking simultaneous on/off cases (SOF).

For the "3-bin" model number of rows, columns, and non-zero elements were 82938, 74993, and 402296 respectively. Where it was 64182, 58716, and 323461 for the proposed model. The difference in the scale was around 20%, however, it had a significant affect on the performance.

2.6 Conclusion

Having identical generators in power system network can cause symmetry in the UC MILP formulation. It cause redundancy in search space therefore slows down solution process, which is usually the case in the real-world UC instances. Our cut generation method coupled with aggregation approach helps to eliminate symmetry without sacrificing feasibility or optimality of the found schedule. We have shown that, the proposed method could help to reduce computational time significantly where the traditional methods took longer time to solve or fail to find the optimal solution.

Table 2.2: Computational Results for Academic Instances

	3-bin	R			R + SOF + CFC				
Instance	Time	SP	FC	Time	SP	FC	SOF	CFC	Time
1	146	38	6	6	24	4	0	142	4
2	(0.016)	80	10	8	60	5	0	284	6
3	207	252	22	20	183	20	3	426	17
4	47	104	17	12	52	5	0	284	5
5	2443	195	34	30	145	17	0	355	18
6	57	198	29	31	184	16	14	426	15
7	37	96	10	8	56	4	0	284	5
8	143	312	54	64	166	13	2	426	21
9	416	282	57	45	90	13	0	426	11
10	(0.015)	210	37	33	203	20	7	426	17
11	326	160	21	14	76	8	4	355	10
12	(0.013)	198	31	22	198	27	0	426	19
13	755	282	35	37	203	28	1	426	20
14	399	180	34	20	180	35	0	355	23
15	153	198	43	41	169	30	5	426	32
16	82	165	31	30	90	8	0	355	12
17	73	270	44	41	239	40	1	426	27
18	63	228	39	33	143	18	7	426	22
19	1040	246	38	29	119	17	1	426	17
20	83	234	48	44	187	36	11	426	32

Table 2.3: Problems Scale for Academic Instances

	3-bin			R			R + SOF + CFC		
Instance	Row	Col	NZ	Row	Col	NZ	Row	Col	NZ
1	3973	4364	26247	655	863	3998	747	863	4546
2	5085	5293	33274	817	979	4538	1001	979	5718
3	6388	6805	38693	1334	1634	7349	1564	1634	8845
4	6276	7083	36112	1163	1418	6446	1301	1418	7310
5	7162	7823	42532	1264	1610	6911	1494	1610	8407
6	7093	8009	38947	1338	1634	7259	1568	1634	8755
7	7147	7978	41741	1366	1634	6995	1472	1634	8567
8	7306	7851	42070	1336	1634	7323	1566	1634	8819
9	7559	8091	44444	1231	1445	6700	1431	1445	8112
10	7696	8576	42885	1340	1634	7271	1570	1634	8767
11	18473	19732	102163	1058	1192	5613	1210	1192	6697
12	21992	24423	118044	1336	1634	7247	1566	1634	8743
13	22245	24151	119919	1338	1634	7259	1568	1634	8755
14	24303	25891	138155	1264	1610	6987	1494	1610	8483
15	23939	26506	128219	1338	1634	7247	1566	1634	8743
16	24187	26703	129789	1366	1634	7163	1520	1634	8679
17	26156	28478	140737	1336	1634	7247	1566	1634	8743
18	26006	28869	140034	1338	1634	7259	1568	1634	8755
19	25998	28808	139917	1366	1634	7331	1566	1634	8743
20	26538	29639	140940	1338	1634	7259	1568	1634	8755

Table 2.4: Computational Results for CAISO Instances

	3-bin	R + SOF + CFC				
Instance	Time	SP	FC	SOF	CFC	Time
2014-09-01 0%	4306	1131	0	651	1631	540
2014-12-01 0%	(0.026)	1307	1	673	1631	1366
2015-03-01 0%	(0.016)	1480	1	566	1631	675
2015-06-01 0%	(0.032)	1123	1	329	1631	1588
2014-09-01 1%	4643	857	1	463	1631	526
2014-12-01 1%	(0.012)	1651	1	791	1631	1152
2015-03-01 1%	6025	2000	3	828	1631	1164
2015-06-01 1%	(0.013)	1236	0	414	1631	1314
2014-09-01 3%	(0.011)	1834	4	1070	1631	715
2014-12-01 3%	(0.012)	1328	0	784	1631	935
2015-03-01 3%	5732	2265	7	1101	1631	1156
2015-06-01 3%	(0.019)	1850	1	658	1631	1890
2014-09-01 5%	3815	1053	5	531	1631	469
2014-12-01 5%	(0.012)	1320	1	594	1631	859
2015-03-01 5%	6041	1443	4	735	1631	1299
2015-06-01 5%	(0.021)	1727	2	583	1631	2064
2014-09-01 40%	(0.032)	1393	10	1445	1631	3345
2014-12-01 40%	(0.042)	1575	10	1725	1631	4570
2015-03-01 40%	(0.012)	1166	4	1012	1631	1301
2015-06-01 40%	(0.026)	1313	8	1327	1631	1773

Chapter 3

Exploiting Symmetry for the Job Sequencing and Tool Switching Problem

3.1 Introduction

Flexible Manufacturing System (FMS) is an automated production method that is developed to increase efficiency and flexibility in production planning. FMS is mainly applied in metal working industries to automate production and in microelectronic companies for memory allocation McGeoch and Sleator, 1991; Tang and Denardo, 1988a.

Tool management plays a critical role for achieving high productivity in FMS Konak et al., 2008. Early researches refer to this class of problem as a Loading problem Eilon and Christofides, 1971; Rupe and Kuo, 1997; Stecke and Solberg, 1981, many recent studies refer to it as Job Sequencing and Tool Switching problem (SSP) Calmels, 2019, 2022; da Silva et al., 2021. In a general SSP, different jobs are processed in a single machine. Each job requires a different set of tools and the machine's tool magazine needs to be adjusted accordingly. It is not possible to process all jobs without switching the tools, because of the limited magazine capacity. Therefore, the jobs need to be ordered in a way that minimizes the total number of tool switch. The sequencing problem (SP) was proven to be $\mathcal{NP}$-hard Crama et al., 1994, therefore SSP is also belong to $\mathcal{NP}$-hard problem class, as it is more general form of SP.

Symmetry in SSP was first studied by Ghiani et al., 2007. They observed that a sequence of jobs and its reverse order sequence requires the same number of tool switches. They used this property to improve the branch-and-bound procedure developed by Laporte et al., 2004. The same symmetry property was used by Ghiani et al., 2010 to improve the branch-and-cut procedure and generate a better lower bound. da Silva et al., 2021 introduced a symmetry-breaking constraint to enforce the job that requires the most number of tools to be processed in the first half of the processing sequence. It helped to partially eliminate the symmetrical property introduced by Ghiani et al., 2007, and reduced the search space.

In this study, we improve symmetry exploitation by introducing new symmetry-breaking cuts. The SSP was reformulated as a job grouping and sequencing as a multi-commodity flow. One source of symmetry is to have multiple jobs with each use different set of tools and the union of these tool sets doesn't exceed the magazine capacity. In this case, any processing order of these jobs would not change the number of switch count and it will lead to symmetrical and non-dominant solutions. This case is prevented by reformulation of the model. Another type of symmetry is caused by having a job that could be processed in multiple position of the processing order, as the tools required for that job is a subset of multiple magazine setups. The symmetry-breaking cuts are added to the model to eliminate this and other symmetry types to reduce the search space and improve the computation time. A computation study was conducted by using the available data in the literature which has 1050 test instances to test the performance of the proposed approach.

3.2 Problem Description

Let $\mathcal{N} = \{1, \ldots, N\}$ represents the set of jobs that need to be processed in a single machine. Each job (i) requires a set of tools (T_i) from the tool set $\mathcal{T} = \{1, \ldots, M\}$ to be loaded into the magazine before processing the job. The machine has tool capacity C, and for feasibility, it is assumed that $|T_i| \leq C$. The objective is to find the sequence of jobs that minimizes the tool number of switches.

The assumptions are the following: (I) the tool sockets are indifferent and each tool requires a single slot, (II) tool changing times are constant and are identical for all tools, (III) the jobs and the required tools for each job are known in advance, (IV) the required tools for any job doesn't exceed the magazine capacity, (V) the tools do not break or wear out, (VI) the first job in the sequence doesn't require any tool switch.

An example of an SSP instance is shown in Table 3.1. The first row shows the jobs and columns represent the required tools for each job (T_i). In this example, there are 8 tools in total and the magazine capacity is 7. Table 3.2 shows the optimal solution of the instance presented in Table 3.1. In the first row, the optimal sequence of the jobs is given. The unloaded tools are underlined and the loaded tools are given in bold. First, the magazine needs to be loaded with the first six tools and the first six jobs can be processed. Then after job 7, tool 1 need to be replaced with 8. The remaining six jobs can be processed without any tool switches. The last row in Table 3.2 shows the number of tools switch before each job. In total, two switches are required to process all jobs.

In fact, the solution presented in Table 3.1 is only one of the alternative solutions among many. From the alternative visualization of the same instance shown in Table 3.3, it can be clearly seen that any permutation of the first or last six jobs within itself would lead to the same results. Switching the order of the first and last six jobs would also not change the total number of switches. In total, for the given instance,

Table 3.1: An example of SSP instance, visualization (a).

Jobs	1	2	3	4	5	6	7	8	9	10	11	12	13
Tools	1	1	1	1	1	1	2	2	2	2	2	2	3
	2	2	2	2	2	3	3	3	3	3	3	4	4
	3	3	3	3	4	4	4	4	4	4	5	5	5
	4	4	4	5	5	5	5	5	5	6	6	6	6
	5	5	6	6	6	6	6	6	7	7	7	7	7
	6	7	7	7	7	7	7	8	8	8	8	8	8

Table 3.2: The optimal solution of SSP instance shown in Table 3.1.

Jobs	1	2	3	4	5	6	7	8	9	10	11	12	13
Tools	1	1	1	1	1	1	1	2	2	2	2	2	2
	2	2	2	2	2	2	2	3	3	3	3	3	3
	3	3	3	3	3	3	3	4	4	4	4	4	4
	4	4	4	4	4	4	4	5	5	5	5	5	5
	5	5	5	5	5	5	5	6	6	6	6	6	6
	6	6	6	6	6	6	6	7	7	7	7	7	7
	7	7	7	7	7	7	7	**8**	8	8	8	8	8
Switch	0	0	0	0	0	0	0	1	0	0	0	0	0

Table 3.3: An example of SSP instance, visualization (b).

	Jobs												
Tools	1	2	3	4	5	6	7	8	9	10	11	12	13
1	•	•	•	•	•	•	○	○	○	○	○	○	○
2	•	•	•	•	•	○	•	•	•	•	•	•	○
3	•	•	•	•	○	•	•	•	•	•	•	○	•
4	•	•	•	○	•	•	•	•	•	•	○	•	•
5	•	•	○	•	•	•	•	•	•	○	•	•	•
6	•	○	•	•	•	•	•	•	○	•	•	•	•
7	○	•	•	•	•	•	•	○	•	•	•	•	•
8	○	○	○	○	○	○	○	•	•	•	•	•	•

Note: If tool t is required for job i (•) symbol is used, otherwise (○) is used.

there are at least $6! \cdot 6! \cdot 2 \cdot 13$ =13,478,400 alternative or non-dominating optimal solutions.

3.3 Literature Review

The SSP was formally introduced by Tang and Denardo, 1988a. They demonstrated an exact method to solve SSP and tightened the constraints for better performance. Upon the slow performance of the exact method, they developed a Keep Tool Needed Soonest (KTNS) algorithm to find minimum tool switches. This algorithm works under the condition that the order of jobs is known. By combining this method with a Greedy Perturbation (GP) heuristic they could solve small instances very fast. However, it was inefficient for large instances.

Following this study different variants of SSP were studied in the literature. In tool size variation non-uniform tool sizes were considered with the relaxed capacity constraints Crama et al., 2007; Hirvikorpi, Salonen, et al., 2006; Matzliach and Tzur, 2000; Rupe and Kuo, 1997. Multiple machines variation specially focused on parallel machines with uniform or non-uniform magazine capacities Fathi and Barnette, 2002; Gökgür et al., 2018; Özpeynirci et al., 2016; Van Hop and Nagarur, 2004. Multi-objective variation considers the objective of minimizing tool switching instants and minimizing the number of tool switches simultaneously Baykasoğlu and Ozsoydan, 2017, 2018; Mauergauz, 2017; Solimanpur and Rastgordani, 2012. In tool wear variation tool lifetimes are assumed to be either deterministic Dadashi et al., 2016; Hirvikorpi, Nevalainen, et al., 2006 or stochastic Farughi et al., 2017; Hirvikorpi et al., 2007. In this study, we will focus on uniform SSP, which is the most popular variation of SSP Calmels, 2019. In this variation tool sizes are uniform and sequence-independent set-up times are used Chaves et al., 2016; Paiva and Carvalho, 2017; Schwerdfeger and Boysen, 2017; Tang and Denardo, 1988a.

In terms of solution methods used in the uniform SSP literature, heuristic algorithms were used in the majority of the studies Ahmadi et al., 2018; Al-Fawzan and Al-Sultan, 2003; Amaya et al., 2020; Chaves et al., 2016; Mecler et al., 2021; Paiva and Carvalho, 2017; Salonen et al., 2006; Schwerdfeger and Boysen, 2017. However, these methods do not guarantee the optimality of the solution, and in general, they are focused on finding a better sub-optimal solution in a reasonably short time frame. Therefore, in this study, we focus on finding an efficient exact method and compared it with the existing literature.

In the uniform SPP literature, there are very few studies that proposed an exact method to solve the problem. Following the mentioned work of Tang and Denardo, 1988a, Laporte et al., 2004 proposed a reformulation of SSP that provides a better lower bound compared to Tang 1988. Their model is effective with instances up to 25 jobs.

Some of the studies used branch-and-bound methods to improve the solution process. Karakayalı and Azizoğlu, 2006 proposed a branch-and-bound algorithm that is refined by precedence relations and combined with lower and upper-bound improvement techniques. Ghiani et al., 2007 explored symmetrical property of SSP to improve branch-and-bound algorithm developed by Laporte et al., 2004. The same authors proposed a reformulation of SSP as a nonlinear least cost hamiltonian cycle problem and generated a branch-and-cut algorithm Ghiani et al., 2010. This method was able to solve several instances with 45 jobs that couldn't be solved by the previous methods presented in the literature.

More recent studies focused on developing a tighter formulation of SSP. Catanzaro et al., 2015 proposed three MIP models that were tighter than the model proposed by Laporte et al., 2004. They showed that their approach improved the lower bound, compared to the previous studies. However, their models were unable to solve large instances, due to an increased number of variables and valid constraints. da Silva et al., 2021 proposed a multi-commodity flow model presented for SSP. Their approach helped to improve Linear Programming (LP) relaxation and therefore has a better lower bound compared to previous studies.

Please note that minimizing the number of tool switching instants problem (TSIP) is different than SSP. This class of problem also is referred to as the job grouping problem in the earlier literature Ham, 1985; Kusiak, 1986; Tang and Denardo, 1988b. TSIP is a problem in P, and a polynomial algorithm for TSIP was developed by Adjiashvili et al., 2015. The branch-and-bound based algorithm developed by Furrer and Mütze, 2017 could solve TSIP with 2650 jobs in less than 1 second.

Contrary to TSIP, SSP is an $\mathcal{NP}$-hard problem Crama et al., 1994; Karakayalı and Azizoğlu, 2006. The exact formulations used for solving SSP are not efficient in terms of solution time. There is a still need for improving existing exact methods and tighter formulations Calmels, 2019.

3.4 New MILP reformulation for the SSP

3.4.1 Job Grouping and Sequencing

Let $B = \{1, \ldots, K\}$ represents a set of job groups or bins where the jobs are clustered. First, the main decision variables and constraints of the proposed *JGS* model are presented. Then, *Symmetry Breaking Cuts* and *Performance Improvement Cuts*, which are added to the *JGS* model to reduce the search space and speed up the solution process are shown.

Decision Variables:

x_{ik} Binary, 1 if job i is placed at bin k; 0 otherwise.

y_{tk} Binary, 1 if tool t is in bin k; 0 otherwise.

v_{tk} Binary, 1 if tool t is loaded to bin k; 0 otherwise.

w_{tk} Binary, 1 if tool t is removed from bin k; 0 otherwise.

JGS Model

$$\min \quad \sum_{t=1}^{T} \sum_{k=2}^{K} v_{tk} \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{B}} x_{ik} = 1 \qquad \forall i \in \mathcal{N} \tag{3.1b}$$

$$\sum_{t \in \mathcal{T}} y_{tk} = C \qquad \forall k \in \mathcal{B} \tag{3.1c}$$

$$x_{ik} \leq y_{tk} \qquad \forall t \in T_i, \forall k \in \mathcal{B}, \forall i \in \mathcal{N} \tag{3.1d}$$

$$y_{tk} = y_{t(k-1)} + v_{tk} - w_{tk} \qquad \forall t \in \mathcal{T}, \forall k \in \mathcal{B}. \tag{3.1e}$$

Here, the objective function (3.1a) minimizes the number of tool switches starting from the second bin. Constraint (3.1b) ensures every job is placed in one of the bins, Constraint (3.1c) ensures number of the tools in the magazine is equal to the capacity, Constraint (3.1d) ensures that if a job placed in a bin then all necessary tools required for that job are available, Constraint (3.1e) is used for ensuring the relationship between loaded, unloaded and existing tools.

Symmetry Breaking Cuts

If a job could be done earlier then it shouldn't be delayed. The job need to be processed as soon as all necessary tools are available. Constraint (1f) ensures this condition:

$$x_{ik} \leq \sum_{t \in T_i} v_{tk} \qquad \forall k \in \mathcal{B}, \forall i \in \mathcal{N}. \tag{1f}$$

If no tool is added to a bin then that bin is redundant. Constraint (1g) ensures that redundant bins are placed towards the last bins:

$$C \sum_{t} v_{tk} \geq \sum_{t} v_{t(k+1)} \qquad \forall k \in \mathcal{B}. \tag{1g}$$

If no tool is removed from a bin, then it means that the following bin is redundant. Constraint (1h) ensures that, if no tool is removed from a bin at position k then don't remove any tool from later bins:

$$C \sum_{t} w_{tk} \geq \sum_{t} w_{t(k+1)} \qquad \forall k \in \mathcal{B}. \tag{1h}$$

Constraint (1i) ensures that, if a tool could be removed from the magazine then it is removed as soon as possible:

$$\sum_{(i|\, t \in T_i)} x_{i(k-1)} \geq w_{tk} \qquad \forall t \in \mathcal{T}, \forall k \in \mathcal{B}. \tag{1i}$$

If job i is not done until bin k and at bin k all necessary tools for job i are available, then Constraint (1j) ensures that job i is placed at bin k:

$$x_{ik} \geq 1 - \sum_{b=1}^{k-1} x_{ib} + \sum_{t \in T_i} y_{tk} - |T_i| \qquad \forall k \in \mathcal{B}, \forall i \in \mathcal{N}. \tag{1j}$$

Constraint (1j) could be also written alternative, as:

$$\sum_{b=1}^{k} x_{ib} \geq \sum_{t \in T_i} y_{tk} - |T_i| + 1 \qquad \forall k \in \mathcal{B}, \forall i \in \mathcal{N}. \tag{1k}$$

Performance Improvement Cuts

Assume that a binary decision variable z_k is equal to 1 if, there is at least one job placed at bin k, 0 otherwise. Two more constraints are required for linking this new decision variable to the rest of the model:

$$z_k \geq x_{ik} \qquad \forall k \in \mathcal{B}, \forall i \in \mathcal{N} \tag{1l}$$

$$z_k \leq \sum_{t} v_{tk} \qquad \forall k \in \mathcal{B}. \tag{1m}$$

Constraint (1l) makes sure that if there is at least one job in bin k, then bin k should be available, Constraint (1m) ensures that if no tools are added to bin k, then bin k is not available. After taking care of the new decision variable, the following constraints could be considered for tightening the proposed formulation:

$$z_k \geq z_{k+1} \qquad \forall k \in \mathcal{B}. \tag{1n}$$

Constraint (1n) ensures that, if a bin is not utilized then later bins are also not unitized. This condition will prevent empty bins to be scattered among the bins, instead, they will be accumulated towards the end of the order.

The next two cuts are generated by using graph theory. Assume that, $G = (V, E)$ has $V = \mathcal{N}$ and edges between vertices if two jobs couldn't be placed in the same bin. The limited magazine capacity doesn't allow job i and j to be placed into the same bin if the $|T_i \cup T_j| > C$ condition is satisfied. Assume that $\mathcal{L} = \{1, \ldots, L\}$ represents set of cliques, and O_l represents set of jobs in the clique l, where $l \in \mathcal{L}$, and $P = \{1, \ldots, \max_l \{O_l\}\}$. By using these notations the following cuts could be defined:

$$z_k \geq \sum_{i \in O_l} x_{ik} \qquad \forall l \in \mathcal{L}, \forall k \in \mathcal{B} \tag{1o}$$

$$z_k = 1 \qquad \forall k \in P. \tag{1p}$$

Constraint (1o) ensures that the jobs that form a clique in graph G are not placed in the same bin. Constraint (1p) ensures that the number of available bins is at least equal to maximal clique size, and the available bins are positioned first among all bins. All decision variables in this model are set to binary:

$$x_{ik}, y_{tk}, v_{tk}, w_{tk}, z_k \in \{0, 1\} \qquad \forall t \in \mathcal{T}, \forall k \in \mathcal{B}, \forall i \in \mathcal{N}. \tag{1q}$$

3.4.2 Job Grouping and Sequencing as a Multi-commodity Flow

The symmetry-breaking and tightening cuts in *JGS* model improves the upper-bound by reducing the search space. This model could be further improved by adopting a multi-commodity flow approach developed by da Silva et al., 2021, which will help to have better lower-bound.

Consider the graph in Figure 3.1. Here, the blue nodes S,R,D represent starting tools, tool repository, and detached tools, respectively. The red nodes are used for bins and the green nodes for jobs. The straight arrows represent the flow of tools, and the dashed arrows show the possible placement of jobs ($\mathcal{N}$) into bins ($\mathcal{B}$). A tool sent to the repository (R) could be re-used, however, detached tools (D) can not be used again.

Assume that a binary decision variable f_{abt} is equal to 1, if tool t is transferred from node a to b, where $a \in \{S, R, \mathcal{B}\}$ and $b \in \{R, D, \mathcal{B},\}$; 0 otherwise. The new Job Grouping and Sequencing as a Multi-commodity Flow model could be written as follows:

JGSMF Model

$$\min \quad \sum_{t\in\mathcal{T}}\sum_{k=1}^{K-1} f_{kDt} + \sum_{t\in\mathcal{T}}\sum_{k=1}^{K-2} f_{kRt} \tag{3.2a}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{B}} x_{ik} = 1 \qquad \forall i \in \mathcal{N} \tag{3.2b}$$

$$\sum_{t\in\mathcal{T}} f_{(k-1)kt} = C \qquad \forall k \in \mathcal{B} \tag{3.2c}$$

$$x_{ik} \leq f_{(k-1)kt} \qquad \forall t \in T_i, \forall k \in \mathcal{B}, \forall i \in \mathcal{N} \tag{3.2d}$$

$$f_{S1t} + f_{SRt} = 1 \qquad \forall t \in \mathcal{T} \tag{3.2e}$$

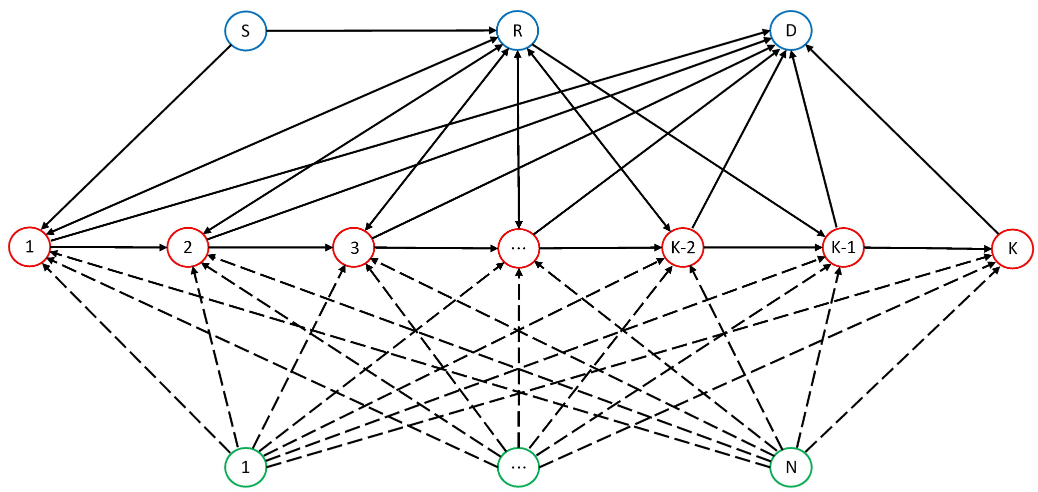

Figure 3.1: Multi-Commodity Flow Graph.

$$\sum_{k \in \mathcal{B}} f_{kDt} = 1 \qquad \forall t \in \mathcal{T} \tag{3.2f}$$

$$f_{(k-1)kt} + f_{Rkt} = f_{k(k+1)t} + f_{kDt} + f_{kDt} \qquad \forall t \in \mathcal{T}, \forall k \in \mathcal{B} \tag{3.2g}$$

$$f_{(K-2)(K-1)t} + f_{R(K-1)t} = f_{(K-1)Kt} + f_{(K-1)Dt} \qquad \forall t \in \mathcal{T} \tag{3.2h}$$

$$f_{(K-1)Kt} = f_{KDt} \qquad \forall t \in \mathcal{T} \tag{3.2i}$$

$$f_{SRt} + \sum_{k=1}^{K-2} f_{kRt} = \sum_{k=1}^{K-1} f_{Rkt} \qquad \forall t \in \mathcal{T} \tag{3.2j}$$

$$x_{ik} \leq \sum_{t \in T_i} f_{R(k-1)t} \qquad \forall k \in \mathcal{B}, \forall i \in \mathcal{N} \tag{3.2k}$$

$$C \sum_{t \in \mathcal{T}} f_{R(k-1)t} \geq \sum_{t \in \mathcal{T}} f_{Rkt} \qquad \forall k \in \mathcal{B} \tag{3.2l}$$

$$C \sum_{t \in \mathcal{T}} f_{(k-1)Dt} \geq \sum_{t \in \mathcal{T}} f_{kDt} \qquad \forall k \in \mathcal{B} \tag{3.2m}$$

$$\sum_{(i \mid t \in T_i)} x_{i(k-1)} \geq f_{(k-1)Dt} \qquad \forall t \in \mathcal{T}, \forall k \in \mathcal{B} \tag{3.2n}$$

$$\sum_{b=1}^{k} x_{ib} \geq \sum_{t \in T_i} f_{(k-1)kt} - |T_i| + 1 \qquad \forall k \in \mathcal{B}, \forall i \in \mathcal{N} \tag{3.2o}$$

$$z_k \geq x_{ik} \qquad \forall k \in \mathcal{B}, \forall i \in \mathcal{N} \tag{3.2p}$$

$$z_k \leq \sum_{t \in \mathcal{T}} f_{(k-1)Dt} \qquad \forall k \in \mathcal{B} \tag{3.2q}$$

$$z_k \geq z_{k+1} \qquad \forall k \in \mathcal{B} \tag{3.2r}$$

$$z_k \geq \sum_{i \in O_l} x_{ik} \qquad \forall l \in \mathcal{L}, \forall k \in \mathcal{B} \tag{3.2s}$$

$$z_k = 1 \qquad \forall k \in P \tag{3.2t}$$

$$x_{ik}, z_k, f_{abt} \in \{0, 1\} \qquad \forall (a, b) \in \{S, R, D, \mathcal{B}\}, \quad \forall t \in \mathcal{T}, \forall k \in \mathcal{B}, \forall i \in \mathcal{N}. \tag{3.2u}$$

Here, the objective function (3.2a) minimizes the flow on arcs (k, D) and (k, R), $k \in \mathcal{B}$, which corresponds to tool switches. Constraint (3.2b) ensures that every job is placed into one of the bins, Constraint (3.2c) enforces the capacity limit, Constraint (3.2d) makes sure that if a job placed into a bin then all necessary tools are available. Constraint (3.2e) enforces that all the tools are available at the start, similarly, Constraint (3.2f) enforces that all tools are removed at the end. Constraint (3.2g), (3.2h), (3.2i), (3.2j) are used for ensuring the flow balance in intermediate nodes. Similar to the *JGS* model, Constraints (3.2k), (3.2l), (3.2m), (3.2n), (3.2o) are used as symmetry-breaking cuts, and Constraints (3.2p), (3.2q), (3.2r), (3.2s) are used for tightening the formulation. Finally, Constraint (3.2u) enforces that all decision variables are binary.

3.4.3 Solution Procedure

The *JGSMF* model has $K(3M+N+K+5)-2$ decision variables and $K(MN+3N+2M+L+5)+5M+N+\max_l\{O_l\}$ constraints. In this formulation, all parameters are fixed, except the number of bins (K), which affects problem size significantly. Having N bins ($K=N$) would be sufficient to solve the problem, but it will be computationally difficult. On the other hand, having very few bins may lead to an infeasible solution because of the capacity constraint, or can cause to have an infeasible or non-optimal solution.

The proposed solution has three phases (P_1, P_2, P_3). In first phase (P_1) B_1 bins are used in *JGSMF* model and solve it to optimality to find an initial solution. In the beginning B_1 is set to B_0, where $B_0 = max(5, \min[\max_l\{O_l\}, N])$. The number of optimal switches (S_1) gives an upper-bound (UB) to the SSP. If it is infeasible, B_1 is incremented by 2 and solved repetitively until the feasible solution is found or the time limit (T_1) is reached.

In second phase (P_2), number of bins are set to B_2, where $B_2 = B_1+1$. If optimal switches in the second phase (S_2) is equal to S_1 then it is terminated, otherwise, UB is updated. The problem is resolved by incrementing B_2 by 1 and corresponding S_2 is found. The process continued until the time limit (T_2) is reached or there is no reduction in optimal switch (S_2) in comparison to the previous run.

In the third phase (P_3), additional constraints are added to the *JGSMF* model to fix the flow between each bin and all bins are fixed to be open. Constraint (3.3a) and (3.3b) are used to fix the flow to 1, Constraint (3.3c) is used to set all bins to be open:

$$\sum_{t \in \mathcal{T}} f_{kRt} + f_{kDt} = 1 \qquad \forall k \in \{1, \ldots, K-2\} \tag{3.3a}$$

$$\sum_{t \in \mathcal{T}} f_{(K-1)Dt} = 1 \tag{3.3b}$$

$$z_k = 1 \qquad \forall k \in \mathcal{B}. \tag{3.3c}$$

If it is known that S switch is a feasible answer to an SSP problem, then $S+1$ bins should be sufficient enough to solve *JGSMF* model. At the end of phase two, it is clear that S_2+1 bins would be sufficient enough. Therefore, the bins in P_3 are fixed to B_3, where $B_3 = S_2$, to test the feasibility of solving the problem with 1 less switch. If it is infeasible, then it means that B_3 switches are optimal, otherwise B_3 is decremented by 1, UB is updated, and the problem is solved again. P_3 is terminated if an infeasible solution is found or the time limit (T_3) is reached. The solution procedure is described in Figure 3.2. Here, $\{t_1, t_2, t_3\}$ represents solution time of an instance of procedure $\{P_1, P_2, P_3\}$ respectively.

Among the three phases, P_1 uses the lowest number of bins, therefore it is the fastest. The time limit of P_1 is set to the maximum given time (1 hour). If a test instance is not solved within this time then it is reported as unsolved, the remaining time is divided equally between P_2 and P_3.

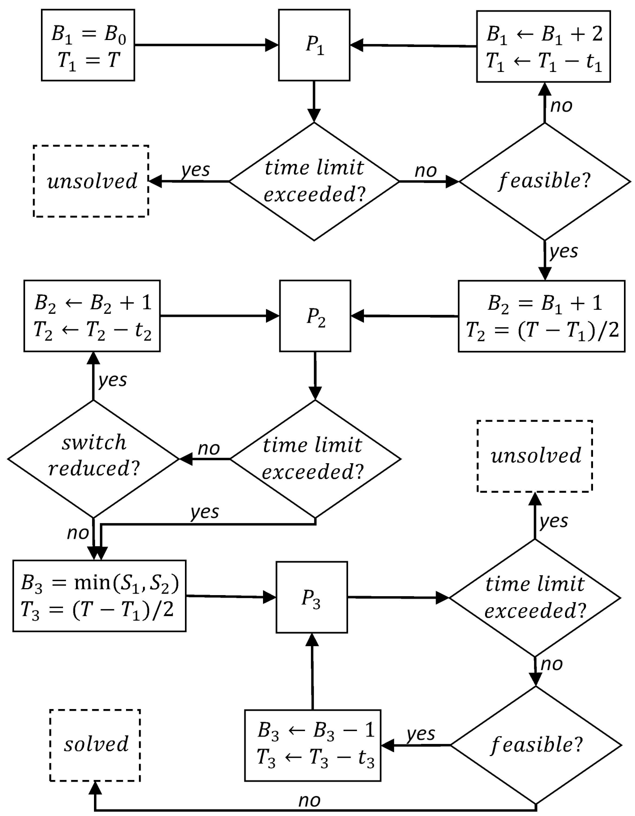

Figure 3.2: The Flow of Solution Process.

3.5 Computational Analysis

3.5.1 Comparison Overview

We demonstrate a performance comparison of the proposed reformulation supported by the solution process described in Figure 3.2. From now on, we will be referring to the proposed approach as JGSMF. It will be compared with the fastest known method in the literature (SSPMF) developed by da Silva et al., 2021.

All computational experiments were performed on a server with AMD Ryzen Threadripper 2950X 16-Core processor and 64 GB RAM running Ubuntu 18.04.6 LTS. Gurobi 8.0.1 with Python interface is used for solving the the MIP problems. The solution time is limited to 3600 seconds. The remaining Gurobi parameters were set to default for solving all instances. If an instance is not solved within the given time the optimality gap is reported.

3.5.2 Data Set Description

The dataset presented by Yanasse et al., 2009 was used to test the effectiveness of the proposed approach. In this database, there are five clusters, and in each cluster, there are 8-15 groups, and in each group, there are 10-130 instances. Among the five clusters, the one with the fewest number of jobs is eliminated as it could be easily solved by the two compared methods and the previous methods presented in the literature. The resulting data set has 1050 instances in total. These test instances could be accessed at (https://sites.google.com/site/antoniochaves/publications/data).

There is a condition that could improve the solution time of a test instance. If a set of required tools for job i is a subset of required tools of job j ($T_i \subseteq T_j$), then job j is referred as *dominating* job. In such a case, all *dominated* jobs could be emitted from the job set ($\mathcal{N}$) prior to the modeling in order to reduce the complexity of the problem. Then, after finding the optimal job processing sequence they could be added to the sequence by placing it right after the *dominating* job. It is worth mentioning that, none of the tested instances has such a condition.

3.5.3 Performance Comparison

The comparison was made first by comparing the number of instances solved to optimality (O) out of I instances. If the number of solved instances are the same then the average run time of solved instances (R) are compared. The superior result is presented in bold font. If some of the instances could not be solved in a given time limit then optimality gaps of commonly unsolved instances (G) were reported.

The size of the problem in each test cluster and an overview of the computational study is presented in Table 3.4. Here, the range of jobs (N), required tools (M), and magazine capacity (C) are given in respective order. The total number of instances in each group is represented by *TI*, and *TO* shows the total number of optimally solved instances.

Both methods were able to solve all instances in data cluster A. On average, the SSPMF method is faster than JGSMF method, see Table 3.5. However, this difference is usually few seconds where almost all instances were solved under a minute. JGSMF solves the model in each phase (P_1, P_2, P_3), therefore, solution speed slowed down for the easy instances.

As the problem size gets slightly increased JGSMF was able to solve more test instances, see Table 3.6. Approximately 16% more instances could be solved with JGSMF and the average solution time is faster for these solved instances for data cluster B. In terms of the average optimality gap for commonly unsolvable cases, JGSMF is slightly (1.4%) better than SSPMF. In comparison to data cluster A, the run times difference is getting significantly bigger for data cluster B, see Figure 3.3. The yellow and green coloured box plots in Figure 3.3, 3.4, 3.5 represents SSPMF and JGSMF, respectively.

Table 3.4: Overview of the problem size and number of solved instances.

Cluster	N Range	M Range	C Range	TI	SSPMF TO	JGSMF TO
A	9-9	15-25	5-20	370	370	370
					difference:	**+0%**
B	15-15	15-25	5-20	340	227	263
					difference:	**+16%**
C	20-25	15-25	5-20	260	114	169
					difference:	**+33%**
D	10-15	10-20	4-12	80	71	75
					difference:	**+5%**

Table 3.5: Computational results for the data cluster A.

					SSPMF			JGSMF		
Group	N	M	C	I	O	R	G	O	R	G
A1	9	15	5	10	10	**4.7**	-	10	11.8	-
A2	9	15	10	30	30	**3.0**	-	30	6.9	-
A3	9	20	10	20	20	**8.7**	-	20	31.4	-
A4	9	20	15	50	50	**4.6**	-	50	11.0	-
A5	9	25	10	20	20	**12.5**	-	20	73.2	-
A6	9	25	15	40	40	**9.6**	-	40	45.8	-
A7	9	25	20	130	130	**5.0**	-	130	16.1	-
A8	9	20	5	10	10	**6.2**	-	10	14.4	-
A9	9	20	10	10	10	0.1	-	10	0.1	-
A10	9	20	15	10	10	0.2	-	10	**0.04**	-
A11	9	25	5	10	10	**1.9**	-	10	3.3	-
A12	9	25	10	10	10	0.1	-	10	0.1	-
A13	9	25	15	10	10	**0.03**	-	10	0.1	-
A14	9	25	20	10	10	**0.02**	-	10	0.1	-

Table 3.6: Computational results for the data cluster B.

					SSPMF			JGSMF		
Group	N	M	C	I	O	R	G	O	R	G
B1	15	15	5	10	7	579	0.12	**9**	316	0.12
B2	15	15	10	30	25	470	-	**30**	284	-
B3	15	20	5	10	7	1107	0.05	**8**	527	0.05
B4	15	20	10	30	22	957	0.19	22	**661**	0.11
B5	15	20	15	60	44	949	0.20	**50**	463	0.19
B6	15	25	5	10	**5**	1934	0.14	3	1639	0.16
B7	15	25	10	30	16	802	0.24	**19**	827	0.34
B8	15	25	15	60	30	952	0.26	**40**	540	0.23
B9	15	25	20	100	71	584	0.24	**82**	452	0.22

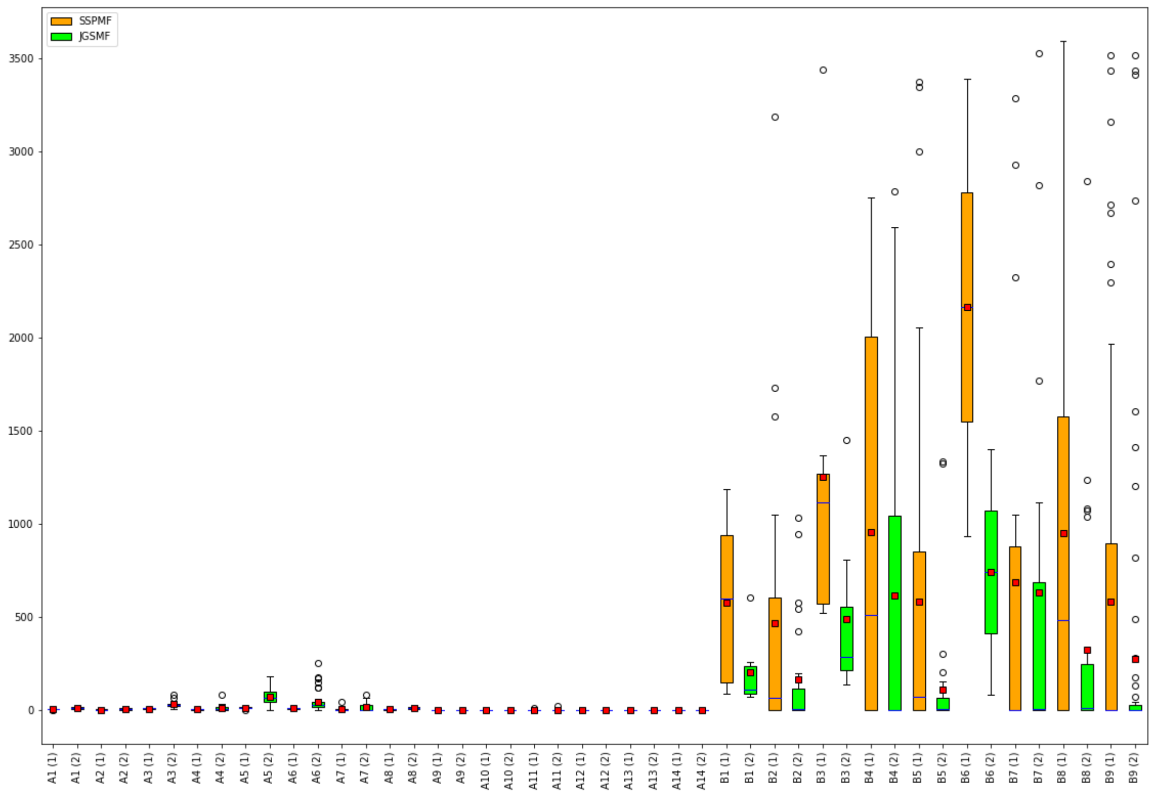

Figure 3.3: Comparison of the solution times for the instances in cluster A and B, that are solved with both methods.

In data cluster C, 33% more instances are solvable by JGSMF, see Table 3.7. The average run time of the JGSMF is higher as it additionally contains the run time of the instance that is not solvable by SSPMF. The optimality gap for commonly unsolvable cases with JGSMF is 39% less than SSPMF.

The performance differences for data cluster D are almost identical where only four (5%) more instances were solved by JGSMF. The run time distribution of commonly solved instances in Figure 3.4 shows that majority of run times are close to zero for data clusters C and D. In one test case (C7) the average run time of commonly solved instances by SSPMF is significantly higher than JGSMF, however, in the rest of the cases their performance is either very close or JGSMF performed better.

In general, there are few instances in that JGSMF is outperformed by SSPMF. The optimality gap of commonly unsolved instances shows a similar result, see Figure 3.5. By analyzing these instances it was seen that the higher run time of JGSMF was caused by assigning a small number to the initial bin count (B_0). Therefore, more time was spent to complete P_1 to get a feasible bin count, and there was little or no time left for processing P_3.

3.6 Conclusion

In this research, we studied an $\mathcal{NP}$-hard combinatorial optimization problem arising from manufacturing systems, the Job Sequence and Switching Problem. A new reformulation was proposed to model SSP as a Job Grouping and Sequencing with a Multi-commodity Flow. The symmetrical property of SSP was exploited and symmetry-breaking cuts were added to the model to reduce the search space and speed up the solution process. The formulation is tightened further by converting SSP to an undirected graph and using maximum clique size as an initial bin count. To get the maximum performance from the proposed reformulation the solution procedure is divided into three phases and in each phase, the model was solved iteratively until the optimality of the solution was guaranteed.

Table 3.7: Computational results for the data cluster C.

					SSPMF			JGSMF		
Group	N	M	C	I	O	R	G	O	R	G
C1	20	15	5	10	0	-	0.44	**1**	802	0.26
C2	20	15	10	20	10	138	0.50	**17**	412	0.08
C3	20	20	5	10	0	-	0.35	**1**	921	0.27
C4	20	20	10	10	6	33	0.14	**7**	12	0.08
C5	20	20	15	30	12	71	-	**30**	381	-
C6	20	25	5	10	0	-	0.36	**1**	2487	0.32
C7	20	25	10	10	6	74	0.06	**8**	350	0.06
C8	20	25	15	40	10	1	0.44	10	**0.2**	0.33
C9	20	25	20	40	26	4	0.41	**37**	83	0.12
C10	25	15	10	10	8	993	-	**10**	215	-
C11	25	20	10	10	1	19	0.21	**2**	487	0.13
C12	25	20	15	10	1	3485	0.39	**7**	2010	0.24
C13	25	25	10	10	**3**	130	0.19	1	9	0.15
C14	25	25	15	10	10	7	-	10	**1**	-
C15	25	25	20	30	21	8	0.40	**27**	3	0.13

Table 3.8: Computational results for the data cluster D.

					SSPMF			JGSMF		
Group	N	M	C	I	O	R	G	O	R	G
D1	10	10	4	10	10	**6**	-	10	10	-
D2	10	10	5	10	10	4	-	10	4	-
D3	10	10	6	10	10	0.4	-	10	**0.2**	-
D4	10	10	7	10	10	0.04	-	10	0.04	-
D5	15	20	6	10	8	1255	0.25	8	**840**	0.25
D6	15	20	8	10	5	444	0.20	**7**	4	0.06
D7	15	20	10	10	9	4	-	**10**	4	-
D8	15	20	12	10	9	98	-	**10**	8	-

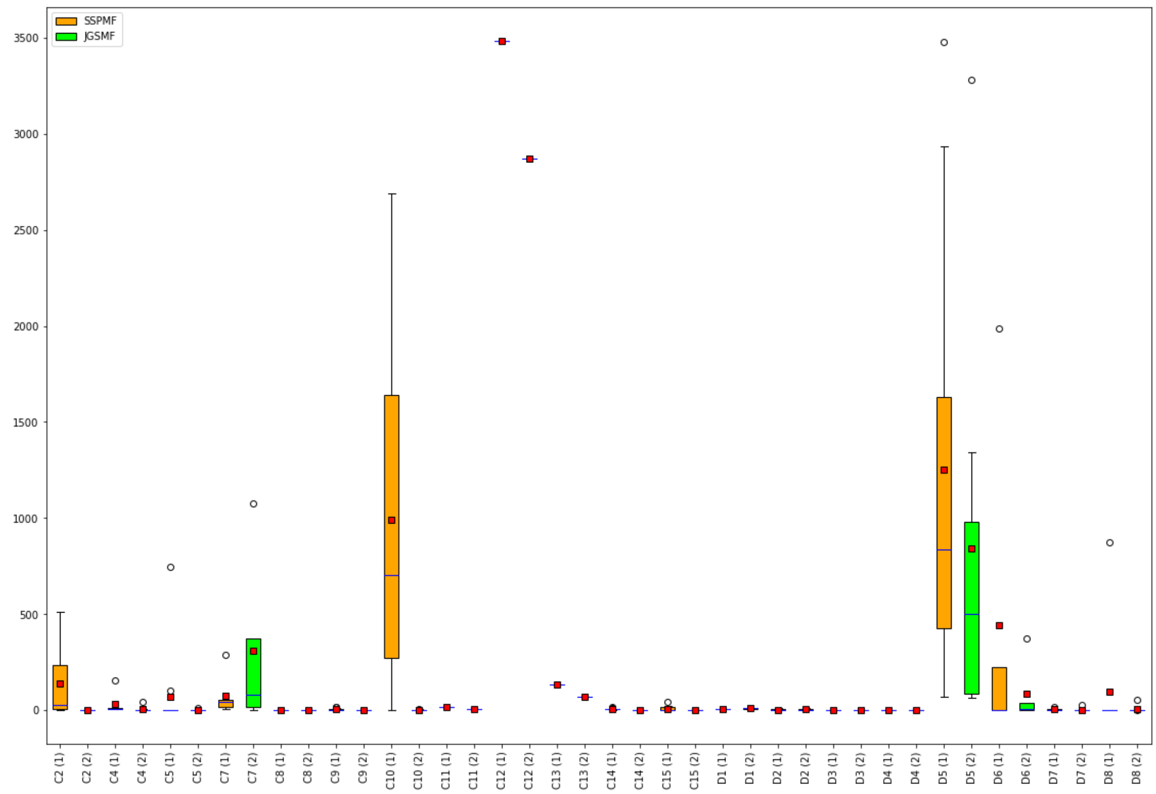

Figure 3.4: Comparison of the solution times for the instances in cluster C and D, that are solved with both methods.

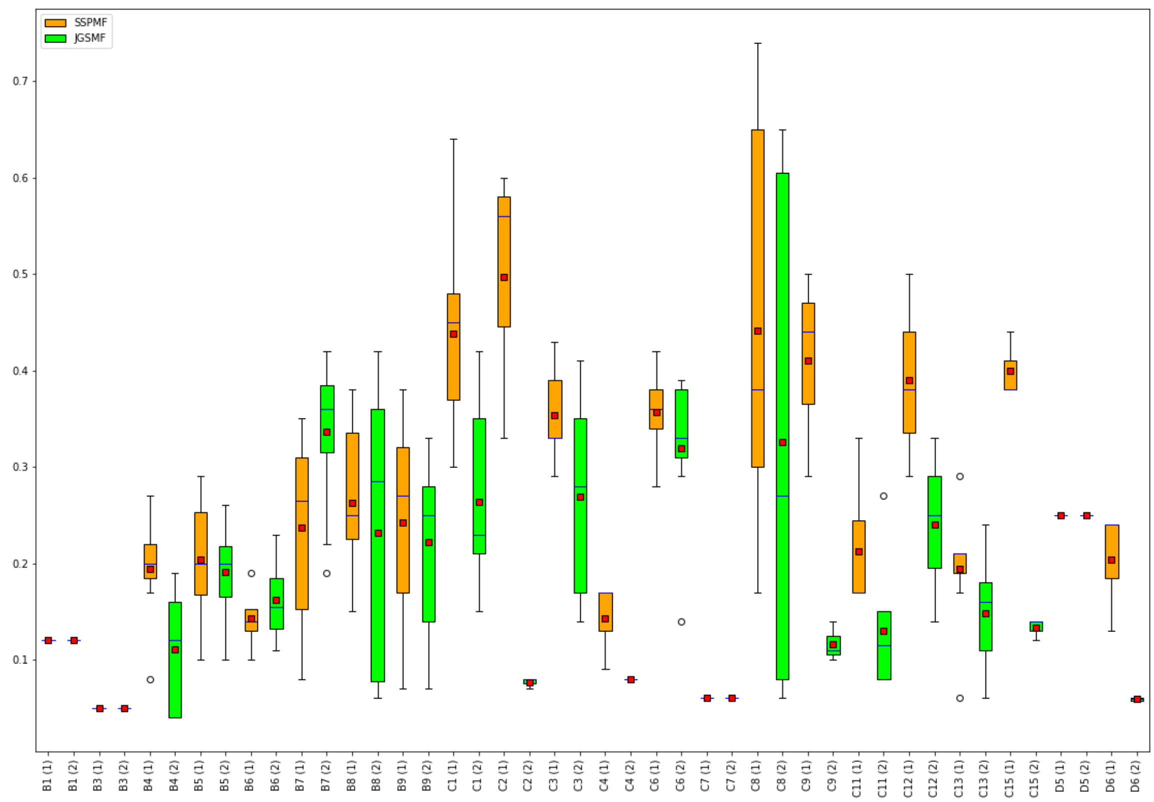

Figure 3.5: Comparison of the optimality gaps for the instances that couldn't be solved with either methods.

The conducted computational study showed that as the size of the problem grows the proposed method was able to solve up to 33% more instances compared to the state-of-the-art method. The optimality gap was reduced up to 39% for the large instances that are not solvable within the given time threshold.

Bibliography

Adjiashvili, D., Bosio, S., & Zemmer, K. (2015). Minimizing the number of switch instances on a flexible machine in polynomial time. *Operations Research Letters*, *43*(3), 317–322.

Ahmadi, E., Goldengorin, B., Süer, G. A., & Mosadegh, H. (2018). A hybrid method of 2-tsp and novel learning-based ga for job sequencing and tool switching problem. *Applied Soft Computing*, *65*, 214–229.

Alemany, J., Magnago, F., Moitre, D., & Pinto, H. (2014). Symmetry issues in mixed integer programming based unit commitment. *International Journal of Electrical Power & Energy Systems*, *54*, 86–90.

Alemany, J., Komarnicki, P., Lin, J., & Magnago, F. (2016). Exploiting symmetry in unit commitment solutions for a large-scale electricity market. *Electric Power Systems Research*, *140*, 363–377.

Al-Fawzan, M., & Al-Sultan, K. (2003). A tabu search based algorithm for minimizing the number of tool switches on a flexible machine. *Computers & industrial engineering*, *44*(1), 35–47.

Amaya, J. E., Cotta, C., Fernández-Leiva, A. J., & Garcıa-Sánchez, P. (2020). Deep memetic models for combinatorial optimization problems: Application to the tool switching problem. *Memetic Computing*, *12*(1), 3–22.

An, Y., & Zeng, B. (2014). Exploring the modeling capacity of two-stage robust optimization: Variants of robust unit commitment model. *IEEE transactions on Power Systems*, *30*(1), 109–122.

Baum, S., & Trotter, L. E. (1978). Integer rounding and polyhedral decomposition for totally unimodular systems. In *Optimization and operations research* (pp. 15–23). Springer.

Baykasoğlu, A., & Ozsoydan, F. B. (2017). Minimizing tool switching and indexing times with tool duplications in automatic machines. *The International Journal of Advanced Manufacturing Technology*, *89*(5), 1775–1789.

Baykasoğlu, A., & Ozsoydan, F. B. (2018). Minimisation of non-machining times in operating automatic tool changers of machine tools under dynamic operating conditions. *International Journal of Production Research*, *56*(4), 1548–1564.

Bendotti, P., Fouilhoux, P., & Rottner, C. (2017). On the complexity of the unit commitment problem. optimization online.

Bertsimas, D., Litvinov, E., Sun, X. A., Zhao, J., & Zheng, T. (2012). Adaptive robust optimization for the security constrained unit commitment problem. *IEEE transactions on power systems*, *28*(1), 52–63.

Bertsimas, D., & Tsitsiklis, J. N. (1997). *Introduction to linear optimization* (Vol. 6). Athena Scientific Belmont, MA.

Bhardwaj, A., Kamboj, V. K., Shukla, V. K., Singh, B., & Khurana, P. (2012). Unit commitment in electrical power system-a literature review. *2012 IEEE international power engineering and optimization conference Melaka, Malaysia*, 275–280.

Borghetti, A., Frangioni, A., Lacalandra, F., Lodi, A., Martello, S., Nucci, C., & Trebbi, A. (2001). Lagrangian relaxation and tabu search approaches for the unit commitment problem. *2001 IEEE Porto Power Tech Proceedings (Cat. No. 01EX502)*, *3*, 7–pp.

Borghetti, A., Frangioni, A., Lacalandra, F., & Nucci, C. A. (2003). Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. *IEEE Transactions on Power Systems*, *18*(1), 313–323.

Calmels, D. (2019). The job sequencing and tool switching problem: State-of-the-art literature review, classification, and trends. *International Journal of Production Research*, *57*(15-16), 5005–5025.

Calmels, D. (2022). An iterated local search procedure for the job sequencing and tool switching problem with non-identical parallel machines. *European Journal of Operational Research*, *297*(1), 66–85.

Carrión, M., & Arroyo, J. M. (2006a). A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems*, *21*(3), 1371–1378. https://doi.org/10.1109/TPWRS.2006.876672

Carrión, M., & Arroyo, J. M. (2006b). A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on power systems*, *21*(3), 1371–1378.

Castillo, A., & Gayme, D. F. (2014). Grid-scale energy storage applications in renewable energy integration: A survey. *Energy Conversion and Management*, *87*, 885–894.

Catanzaro, D., Gouveia, L., & Labbé, M. (2015). Improved integer linear programming formulations for the job sequencing and tool switching problem. *European journal of operational research*, *244*(3), 766–777.

Chao, H.-p. (2019). Incentives for efficient pricing mechanism in markets with non-convexities. *Journal of Regulatory Economics*, *56*(1), 33–58.

Chaves, A. A., Lorena, L. A. N., Senne, E. L. F., & Resende, M. G. (2016). Hybrid method with cs and brkga applied to the minimization of tool switches problem. *Computers & Operations Research*, *67*, 174–183.

Crama, Y., Moonen, L. S., Spieksma, F. C., & Talloen, E. (2007). The tool switching problem revisited. *European Journal of Operational Research*, *182*(2), 952–957.

Crama, Y., Oerlemans, A. G., & Spieksma, F. C. (1994). Minimizing the number of tool switches on a flexible machine. In *Production planning in automated manufacturing* (pp. 165–195). Springer.

Dadashi, H., Moslemi, S., & Mirzazadeh, A. (2016). Optimization of a new tool switching problem in flexible manufacturing systems with a tool life by a genetic algorithm. *International Journal of Industrial and Manufacturing Systems Engineering*, *1*(3), 52.

da Silva, T. T., Chaves, A. A., & Yanasse, H. H. (2021). A new multicommodity flow model for the job sequencing and tool switching problem. *International Journal of Production Research*, *59*(12), 3617–3632.

Denholm, P., & Hand, M. (2011). Grid flexibility and storage required to achieve very high penetration of variable renewable electricity. *Energy Policy*, *39*(3), 1817–1830.

Eilon, S., & Christofides, N. (1971). The loading problem. *Management Science*, *17*(5), 259–268.

Farughi, H., Dolatabadiaa, M., Moradi, V., Karbasi, V., & Mostafayi, S. (2017). Minimizing the number of tool switches in flexible manufacturing cells subject to tools reliability using genetic algorithm. *Journal of Industrial and Systems Engineering*, *10*(special issue on Quality Control and Reliability), 17–33.

Fathi, Y., & Barnette, K. (2002). Heuristic procedures for the parallel machine problem with tool switches. *International Journal of Production Research*, *40*(1), 151–164.

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

Frangioni, A., & Gentile, C. (2006). Perspective cuts for a class of convex 0–1 mixed integer programs. *Mathematical Programming*, *106*(2), 225–236.

Frangioni, A., & Gentile, C. (2009). A computational comparison of reformulations of the perspective relaxation: Socp vs. cutting planes. *Operations Research Letters*, *37*(3), 206–210.

Frangioni, A., Gentile, C., & Lacalandra, F. (2008a). Solving unit commitment problems with general ramp constraints. *International Journal of Electrical Power & Energy Systems*, *30*(5), 316–326.

Frangioni, A., Gentile, C., & Lacalandra, F. (2008b). Tighter approximated milp formulations for unit commitment problems. *IEEE Transactions on Power Systems*, *24*(1), 105–113.

Fu, B., Ouyang, C., Li, C., Wang, J., & Gul, E. (2019). An improved mixed integer linear programming approach based on symmetry diminishing for unit commitment of hybrid power system. *Energies*, *12*(5), 833.

Furrer, M., & Mütze, T. (2017). An algorithmic framework for tool switching problems with multiple objectives. *European Journal of Operational Research*, *259*(3), 1003–1016.

Garcia-Gonzalez, J., de la Muela, R. M. R., Santos, L. M., & Gonzalez, A. M. (2008). Stochastic joint optimization of wind generation and pumped-storage units in an electricity market. *IEEE Transactions on Power Systems*, *23*(2), 460–468.

Gentile, C., Morales-Espana, G., & Ramos, A. (2017). A tight mip formulation of the unit commitment problem with start-up and shut-down constraints. *EURO Journal on Computational Optimization*, *5*(1), 177–201.

Ghiani, G., Grieco, A., & Guerriero, E. (2007). An exact solution to the tlp problem in an nc machine. *Robotics and Computer-Integrated Manufacturing*, *23*(6), 645–649.

Ghiani, G., Grieco, A., & Guerriero, E. (2010). Solving the job sequencing and tool switching problem as a nonlinear least cost hamiltonian cycle problem. *Networks: An International Journal*, *55*(4), 379–385.

Gökgür, B., Hnich, B., & Özpeynirci, S. (2018). Parallel machine scheduling with tool loading: A constraint programming approach. *International Journal of Production Research*, *56*(16), 5541–5557.

Gribik, P. R., Hogan, W. W., Pope, S. L., et al. (2007). Market-clearing electricity prices and energy uplift. *Cambridge, MA*, 1–46.

Ham, I. (1985). An expanding role for group technology. *Computer Integrated Manufacturing Review*, *1*, 21–25.

Herrero, I., Rodilla, P., & Batlle, C. (2018). Enhancing intraday price signals in us iso markets for a better integration of variable energy resources. *The Energy Journal*, *39*(3).

Hirvikorpi, M., Knuutila, T., Leipälä, T., & Nevalainen, O. S. (2007). Job scheduling and management of wearing tools with stochastic tool lifetimes. *International Journal of Flexible Manufacturing Systems*, *19*(4), 443–462.

Hirvikorpi, M., Nevalainen, O., & Knuutila, T. (2006). Job ordering and management of wearing tools. *Engineering optimization*, *38*(2), 227–244.

Hirvikorpi, M., Salonen, K., Knuutila, T., & Nevalainen, O. S. (2006). The general two-level storage management problem: A reconsideration of the ktns-rule. *European journal of operational research*, *171*(1), 189–207.

Hua, B., & Baldick, R. (2016). A convex primal formulation for convex hull pricing. *IEEE Transactions on Power Systems*, *32*(5), 3814–3823.

Jabr, R. (2012). Tight polyhedral approximation for mixed-integer linear programming unit commitment formulations. *IET Generation, Transmission & Distribution*, *6*(11), 1104–1111.

Jiang, R., Wang, J., & Guan, Y. (2011). Robust unit commitment with wind power and pumped storage hydro. *IEEE Transactions on Power Systems*, *27*(2), 800–810.

Karakayalı, İ., & Azizoğlu, M. (2006). Minimizing total flow time on a single flexible machine. *International Journal of Flexible Manufacturing Systems*, *18*(1), 55–73.

Kazarlis, S. A., Bakirtzis, A., & Petridis, V. (1996). A genetic algorithm solution to the unit commitment problem. *IEEE transactions on power systems*, *11*(1), 83–92.

Knueven, B., Ostrowski, J., & Wang, J. (2018). The ramping polytope and cut generation for the unit commitment problem. *INFORMS Journal on Computing*, *30*(4), 739–749.

Knueven, B., Ostrowski, J., & Watson, J.-P. (2017). Exploiting identical generators in unit commitment. *IEEE Transactions on Power Systems*, *33*(4), 4496–4507.

Knueven, B., Ostrowski, J., Castillo, A., & Watson, J.-P. (2019). A computationally efficient algorithm for computing convex hull prices. *SAND2019-10896 J, Sandia National Laboratories, Albuquerque, NM.*

Knueven, B., Ostrowski, J., & Watson, J.-P. (2020a). A novel matching formulation for startup costs in unit commitment. *Mathematical Programming Computation*, *12*(2), 225–248.

Knueven, B., Ostrowski, J., & Watson, J.-P. (2020b). On mixed-integer programming formulations for the unit commitment problem. *INFORMS Journal on Computing*, *32*(4), 857–876.

Konak, A., Kulturel-Konak, S., & Azizoğlu, M. (2008). Minimizing the number of tool switching instants in flexible manufacturing systems. *International journal of production economics*, *116*(2), 298–307.

Kusiak, A. (1986). Application of operational research models and techniques in flexible manufacturing systems. *European Journal of Operational Research*, *24*(3), 336–345.

Laporte, G., Salazar-Gonzalez, J. J., & Semet, F. (2004). Exact algorithms for the job sequencing and tool switching problem. *IIE transactions*, *36*(1), 37–45.

Lima, R. M., & Novais, A. Q. (2016). Symmetry breaking in milp formulations for unit commitment problems. *Computers & Chemical Engineering*, *85*, 162–176.

López, J. Á., Ceciliano-Meza, J. L., Guillén, I., & Gómez, R. N. (2013). A heuristic algorithm to solve the unit commitment problem for real-life large-scale power systems. *International Journal of Electrical Power & Energy Systems*, *49*, 287–295.

Lorca, A., Sun, X. A., Litvinov, E., & Zheng, T. (2016). Multistage adaptive robust optimization for the unit commitment problem. *Operations Research*, *64*(1), 32–51.

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, *1*(14), 281–297.

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

Matzliach, B., & Tzur, M. (2000). Storage management of items in two levels of availability. *European Journal of Operational Research*, *121*(2), 363–379.

Mauergauz, Y. (2017). Job and tool group scheduling for a machining center. *International Journal of Management Science and Engineering Management*, *12*(4), 280–287.

McGeoch, L. A., & Sleator, D. D. (1991). A strongly competitive randomized paging algorithm. *Algorithmica*, *6*(1), 816–825.

Mecler, J., Subramanian, A., & Vidal, T. (2021). A simple and effective hybrid genetic search for the job sequencing and tool switching problem. *Computers & Operations Research*, *127*, 105153.

Morales-España, G., Gentile, C., & Ramos, A. (2015). Tight mip formulations of the power-based unit commitment problem. *OR spectrum*, *37*(4), 929–950.

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

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

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

Ostrowski, J., Linderoth, J., Rossi, F., & Smriglio, S. (2011). Orbital branching. *Mathematical Programming*, *126*(1), 147–178.

Özpeynirci, S., Gökgür, B., & Hnich, B. (2016). Parallel machine scheduling with tool loading. *Applied Mathematical Modelling*, *40*(9-10), 5660–5671.

Paiva, G. S., & Carvalho, M. A. M. (2017). Improved heuristic algorithms for the job sequencing and tool switching problem. *Computers & Operations Research*, *88*, 208–219.

Palmintier, B., & Webster, M. (2011). Impact of unit commitment constraints on generation expansion planning with renewables. *2011 IEEE power and energy society general meeting*, 1–7.

Papavasiliou, A., & Oren, S. S. (2013). Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network. *Operations research*, *61*(3), 578–592.

Quan, H., Srinivasan, D., & Khosravi, A. (2014). Incorporating wind power forecast uncertainties into stochastic unit commitment using neural network-based prediction intervals. *IEEE transactions on neural networks and learning systems*, *26*(9), 2123–2135.

Rupe, J., & Kuo, W. (1997). Solutions to a modified tool loading problem for a single fmm. *International Journal of Production Research*, *35*(8), 2253–2268.

Saber, A. Y., Senjyu, T., Miyagi, T., Urasaki, N., & Funabashi, T. (2006). Fuzzy unit commitment scheduling using absolutely stochastic simulated annealing. *IEEE transactions on Power systems*, *21*(2), 955–964.

Saksornchai, T., Lee, W.-J., Methaprayoon, K., Liao, J. R., & Ross, R. J. (2005). Improve the unit commitment scheduling by using the neural-network-based short-term load forecasting. *IEEE Transactions on Industry Applications*, *41*(1), 169–179.

Salonen, K., Raduly-Baka, C., & Nevalainen, O. S. (2006). A note on the tool switching problem of a flexible machine. *Computers & Industrial Engineering*, *50*(4), 458–465.

Schiro, D. A., Zheng, T., Zhao, F., & Litvinov, E. (2015). Convex hull pricing in electricity markets: Formulation, analysis, and implementation challenges. *IEEE Transactions on Power Systems*, *31*(5), 4068–4075.

Schulze, T., & McKinnon, K. (2016). The value of stochastic programming in day-ahead and intra-day generation unit commitment. *Energy*, *101*, 592–605.

Schwerdfeger, S., & Boysen, N. (2017). Order picking along a crane-supplied pick face: The sku switching problem. *European Journal of Operational Research*, *260*(2), 534–545.

Sen, S., & Kothari, D. (2002). An equivalencing technique for solving the large-scale thermal unit commitment problem. In *The next generation of electric power unit commitment models* (pp. 211–225). Springer.

Shortt, A., & O'Malley, M. (2010). Impact of variable generation in generation resource planning models. *IEEE PES general meeting*, 1–6.

Simopoulos, D. N., Kavatza, S. D., & Vournas, C. D. (2006). Reliability constrained unit commitment using simulated annealing. *IEEE Transactions on Power Systems*, *21*(4), 1699–1706.

Solimanpur, M., & Rastgordani, R. (2012). Minimising tool switching and indexing times by ant colony optimisation in automatic machining centres. *International Journal of Operational Research*, *13*(4), 465–479.

Stecke, K. E., & Solberg, J. J. (1981). Loading and control policies for a flexible manufacturing system. *The International Journal of Production Research*, *19*(5), 481–490.

Tahanan, M., van Ackooij, W., Frangioni, A., & Lacalandra, F. (2015). Large-scale unit commitment under uncertainty. *4OR*, *13*(2), 115–171.

Tang, C. S., & Denardo, E. V. (1988a). Models arising from a flexible manufacturing machine, part i: Minimization of the number of tool switches. *Operations research*, *36*(5), 767–777.

Tang, C. S., & Denardo, E. V. (1988b). Models arising from a flexible manufacturing machine, part ii: Minimization of the number of switching instants. *Operations Research*, *36*(5), 778–784.

Tesfatsion, L. (2020). *A new swing-contract design for wholesale power markets*. John Wiley & Sons.

Van Hop, N., & Nagarur, N. N. (2004). The scheduling problem of pcbs for multiple non-identical parallel machines. *European Journal of Operational Research*, *158*(3), 577–594.

van Ackooij, W., Lopez, I. D., Frangioni, A., Lacalandra, F., & Tahanan, M. (2018). Large-scale unit commitment under uncertainty: An updated literature survey. *Annals of Operations Research*, *271*(1), 11–85.

Viana, A., & Pedroso, J. P. (2013). A new milp-based approach for unit commitment in power production planning. *International Journal of Electrical Power & Energy Systems*, *44*(1), 997–1005.

Victoire, T. A. A., & Jeyakumar, A. E. (2006). A tabu search based hybrid optimization approach for a fuzzy modelled unit commitment problem. *Electric Power Systems Research*, *76*(6-7), 413–425.

Wang, Q., Guan, Y., & Wang, J. (2011). A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output. *IEEE transactions on power systems*, *27*(1), 206–215.

Yanasse, H. H., Rodrigues, R. d. C. M., & Senne, E. L. F. (2009). Um algoritmo enumerativo baseado em ordenamento parcial para resolução do problema de minimização de trocas de ferramentas. *Gestão & Produção*, *16*, 370–381.

Yang, L., Jian, J., Wang, Y., & Dong, Z. (2015). Projected mixed integer programming formulations for unit commitment problem. *International Journal of Electrical Power & Energy Systems*, *68*, 195–202.

Yang, L., Jian, J., Zhu, Y., & Dong, Z. (2014). Tight relaxation method for unit commitment problem using reformulation and lift-and-project. *IEEE Transactions on Power Systems*, *30*(1), 13–23.

Yuan-Kang, W., Chih-Cheng, H., & Chun-Liang, L. (2013). Resolution of the unit commitment problems by using the hybrid taguchi-ant colony system algorithm. *International Journal of Electrical Power & Energy Systems*, *49*, 188–198.

Zhang, Y., Dong, Z. Y., Luo, F., Zheng, Y., Meng, K., & Wong, K. P. (2016). Optimal allocation of battery energy storage systems in distribution networks with high wind power penetration. *IET Renewable Power Generation*, *10*(8), 1105–1113.

Zhao, C., & Guan, Y. (2015). Data-driven stochastic unit commitment for integrating wind generation. *IEEE Transactions on Power Systems*, *31*(4), 2587–2596.

Zheng, Q. P., Wang, J., & Liu, A. L. (2014). Stochastic optimization for unit commitment—a review. *IEEE Transactions on Power Systems*, *30*(4), 1913–1924.

Vita

Najmaddin Akhundov was born in Tovuz, Azerbaijan and attended school there at Nasraddin Tusi Middle School. Later he attended Turkish Private High School in Shirvan city and graduated in 2006.

In 2011, he got his bachelor's degree in Industrial Engineering from Baku Engineering University. After working one year at ENCOTECH consulting company he started his graduate studies in Canada and got a master's degree in Systems Design Engineering from the University of Waterloo in 2015. Upon graduation, he moved back to Azerbaijan and worked as a lecturer at Baku Engineering University for two years. During this time he also gave consulting services to local companies in the area of Operations Research and one of those consulting projects led to a publication in INFORMS journal of Applied Analytics.

In 2017, he continued his graduate studies at the New Jersey Institute of Technology. After one year he transferred the program to the University of Tennessee, Knoxville. His Ph.D. in Industrial Engineering was completed in December 2022.