Improvements for Decomposition Based Methods Utilized in the Development of Multi-Scale Energy Systems

R. Cory Allen[1,2], Funda Iseri[1,2], C. Doga Demirhan[3], Iosif Pappas[4], and Efstratios N. Pistikopoulos[1,2,*]

[1]Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX, USA

[2]Texas A&M Energy Institute, Texas A&M University, College Station, TX, USA

[3]Shell Technology Center, Shell International Exploration and Production Inc., Houston, TX, USA

[4]Shell Technology Center, Shell Global Solutions International B.V., Amsterdam, The Netherlands

[*]Corresponding author: stratos@tamu.edu

Abstract

The optimal design of large-scale energy systems can be found by posing the problem as an integrated multi-period planning and scheduling mathematical programming problem. Due to the complexity of the accompanying mathematical programming problem, decomposition techniques are often required, unfortunately they to are often plagued with converge issues. To address these issues we have derived a set of valid inequalities to strengthen the formulation, and we have developed a machine learning framework to identify which subproblems should be solved in each iteration of the decomposition procedure. We illustrate the effectiveness of the valid inequalities and the machine learning framework through the use of a case study wherein we examine the development of an industrial scale hydrogen-based energy system. The results show that the valid inequalities and the machine learning framework significantly reduce the computational burden of the procedure.

Keywords Mixed-Integer Programming, Benders Decomposition, Valid Inequalities, Integrated Planning and Scheduling, Machine Learning

1 Introduction

Globalization, population growth, and urbanization, in quadem with the liberalization of emerging markets is putting extreme stress on raw materials and rapidly increasing the demands for energy related products [1]. Concurrently, there has been a push towards transitioning from traditional

carbon positive energy systems, which predominantly rely upon fossil fuels, towards towards carbon neutral and carbon negative energy systems [27]. Regrettably, solar arrays and wind turbines, which are the predominant energy generators being utilized in this green transition, are plagued with short-term stochastic disturbances and long-term seasonal fluctuations [5]. This in turn, gives rise to the need for both short-term and long-term energy storage technologies, such as lithium-ion battery banks and hydrogen based dense energy carriers (H-DECs) respectively. These energy storage technologies are utilized to guarantee that consumer energy demands can be met, while simultaneously ensuring that the energy system can remain carbon neutral or carbon negative [8, 20, 22, 21]. To complicate matters even further, there has been a recent uptick in unforeseeable political, social, and natural disruptions to global supply chains [29].

To address these aforementioned challenges, there has been a push towards optimally developed and resilient green energy systems [9, 30]. These energy systems typically operate over multiple locations, produce and consume multiple commodities, and are developed over multiple planning periods [14]. Traditionally this class of problems is posed as an integrated infrastructure planning and operational scheduling problem [18, 24, 2]. Depending on how the dynamics of the processes within the energy system are modeled, the resulting mathematical program may be posed as a mixed-integer linear program (MILP), mixed-integer quadratic program (MIQP), or a mixed-integer non-linear program (MINLP) [19]. It should be noted that some researchers have utilized constraint programming (CP) to model the dynamics of processes and have created a solution approach using both CP and MIP [6].

Unfortunately, integrated infrastructure planning and operational scheduling problems are extremely difficult to solve for real-world sized problems even with "state-of-the-art" mixed-integer programming (MIP) solvers. The difficulty arises due to the multiple temporal and spatial scales that are intrinsic to the structure of the problem [3, 31]. Specifically, there are long term infrastructure planning decisions that are made every few years, medium term tactical decisions that are made every few months, and short term operational decisions that are made every hour. Consequently, there has been various iterative solution strategies put forth in the literature to decompose the integrated infrastructure planning and operational scheduling problem to make it more computationally tractable such as but not limited to Benders decomposition and dual decomposition procedures [16, 18, 28, 15].

In typical Benders decomposition approaches, the infrastructure planning decisions are decoupled from the operational scheduling decisions via a single master problem that captures the infrastructure planning decisions and a set of $|\mathcal{P}| \cdot |\mathcal{W}|$ much smaller subproblems that capture the operational scheduling decisions, where $\mathcal{P}$ is the set of planning periods and $\mathcal{W}$ is the set of scheduling scenarios. To accomplish this, typically integrality must be relaxed in the Benders subproblems; however, it is well known that for this class of problems that the linear programming relaxation of the partially relaxed problem is extremely tight [17, 4]. At each iteration in the Benders decomposition procedure, the Benders master problem is first solved, then the Benders subproblems are solved sequently or in parallel using the optimal infrastructure planning decisions from the master

problem, and finally the Benders cuts are added to the Benders master. This procedure continues until the convergence criteria is reached [16, 28].

Unfortunately, in the Benders decomposition procedure, when the problem is decomposed the temporally dependent material demands for the energy system, the superstructure of the energy system, and the interdependencies between the processes within the energy system is captured in the Benders subproblems, not in the Benders master problem. This small nuance can cause convergence issues, even with the multi-cut Benders decomposition procedure, due to the fact that the Benders master problem only becomes aware these intrinsic issues through the addition of Benders cuts [16, 28].

Moreover, oftentimes the Benders decomposition procedure waste computational resources solving subproblems that are not sensitive to their input parameters. Consequently, when theses subproblems are solved, the corresponding cuts generated marginally affects the convergence gap of the decomposition procedure. This becomes computationally problematic when the subproblems are solved in serial or in parallel if there are more subproblems than numbers of threads.

To alleviate these aforementioned issues, we have a developed a set of valid inequalities and a framework using machine learning to identify which subproblems have a higher probability of affecting the convergence gap. The valid inequalities can be inserted a priori to the Benders decomposition procedure by embedding them directly into the Benders master problem and the machine learning framework can be utilized to select a subset of the subproblems to solve in each iteration that are most probable to improve the convergence gap.

To illustrate the effectiveness of the a priori addition of the valid inequalities and the machine learning framework, we have generated a case study wherein we design a large-scale hydrogen-based energy system utilizing an integrated infrastructure planning and operational scheduling problem. We then conduct a set of computational experiments where we determine the effectiveness of the proposed valid inequalities and the machine learning framework compared to solving the original undecomposed mathematical programming formulation of the problem when it is solved with a "state-of-the-art" mixed-integer programming (MIP) solver: (i) in the first set of experiments, we examine the performance of the multi-cut Benders decomposition procedure; (ii) in the second set of experiments, we examine the performance of the multi-cut Benders decomposition procedure with the a priori addition of the valid inequalities; (iii) in the third set of experiments, we examine the performance of the multi-cut Benders decomposition procedure when the machine learning framework "learns" which subproblems should be solved in every iteration; and (iv) in the final set of experiments, we examine the performance of the multi-cut Benders decomposition procedure when the valid inequalities are added a priori and the machine learning framework "learns" which subproblems should be solved in every iteration.

The outline of the manuscript is as follows: in Section 2, the problem under consideration is formally defined; in Section 3, the Benders decomposition procedure for decomposing the fullspace mathematical programming formulation is presented; in Section 4, the valid inequalities are presented; in Section 5, the machine learning framework utilized to identify' which subproblems should

be solved in each iteration; in Section 6, a case study is presented; in Section 7; the results to a case study wherein we examine the effectiveness of the a priori addition of the valid inequalities and machine learning framework; finally, in Section 8, the closing remarks are given.

2 Problem Definition

Consider a central planner who is trying to develop and/or expand an energy system that spans multiple locations, $\mathcal{L}$, over the course of multiple planning periods, $\mathcal{P}$. The superstructure of this energy system can be mapped to a network representation captured by the graph $G_1(V_1, E_1)$. The vertices in the network representation, V_1, indicate the non-mode based processes, $\mathcal{C}$, mode based processes, $\mathcal{D}$, storage units, $\mathcal{A}$, and transportation mechanisms, $\mathcal{B}$, that transport commodities between locations in the energy system. The directed edges, E_1, in the network representation indicate the pathways in which the commodities can flow between the processes, storage units, and transportation mechanisms.

The central planner would like to harness an integrated multi-period infrastructure planning and operational scheduling approach for determining the optimal development and/or expansion of the energy system. This integrated approach allows the central planner to construct additional components for the energy system, $V_1 \setminus \mathcal{H}$, in the planning periods and to make operational decisions in the scheduling periods, $\mathcal{S}$, during each of the representative scheduling scenarios, $\mathcal{W}$, while simultaneously ensuring that the commodity, $\mathcal{R}$, demands are met, where $\mathcal{H}$ is the set of preexisting components.

The processes in the energy system are divided into two categories: (i) those with simple non-mode process dynamics and variable nameplate capacities; and (ii) those with complex mode based process dynamics and fixed nameplate capacities. The commodities utilized in the energy system, raw materials, intermediate products, and/or final products, can be transported between locations in the energy system via a set of transportation mechanisms, $\mathcal{B}$. The processes and storage units that can be situated at a location, $l \in \mathcal{L}$, in the energy system, and the transportation mechanisms that are able to import or export a commodity to or from that location during a planning period, $p \in \mathcal{P}$, are given by the subset $V_1(p, l) \subseteq V_1$. It should be noted, that to always ensure feasibility, there is a slack commodity source and a slack commodity sink at each location in the energy system.

The central planner has the ability to make the following decisions: (i) determine which components are constructed and what their respective nameplate capacity should be; (ii) determine the set points and possible operating modes of the components in scheduling subproblems; and (iii) determine when and how much of the commodities should be purchased or sold at the slack sources and slack sinks respectively.

3 Benders Decomposition Procedure

In this section, the Multi-cut Benders decomposition procedure is presented to decompose the fullspace MILP formulation of the integrated multi-period infrastructure planning and operational scheduling problem Li et al. [16] – it should be noted that the nomenclature is given at the conclusion and the fullspace MILP formulation is given in the Supplementary Material section of this work.

The Multi-cut Benders decomposition procedure decouples the operational scheduling problems from the infrastructure planning problem, which in turn results into $|\mathcal{P}| \cdot |\mathcal{W}| + 1$ much smaller, seperate problems.

3.1 Benders Master Problem – Infrastructure Planning Problem

The Benders master problem is given by $B_k(\cdot, \cdot, \cdot)$, where: (i) the first input, $\dot{\boldsymbol{v}}$, stores the past k sets of solutions to the infrastructure planning problem; (ii) the second input, $\dot{\boldsymbol{\theta}}$, stores the past k sets of solutions to the operational scheduling problems; and (iii) the third and final input, $\dot{\boldsymbol{\mu}}$, stores the past i sets of solutions to the dual variables that were generated by solving the Benders subproblems.

$$
\begin{aligned}
B_k\big(\dot{\boldsymbol{v}}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\mu}}\big) = \min\ & \sum\nolimits_{n \in V_1 \setminus \mathcal{H}} V_n \cdot v_n + \sum\nolimits_{n \in V_1 \setminus \mathcal{H}} \bar{V}_n \cdot \bar{v}_n + \sum\nolimits_{p \in \mathcal{P}} \sum\nolimits_{w \in \mathcal{W}} \theta_{p,w} \\
\text{s.t. } & \theta_{p,w} \geq \dot{\theta}^{\hat{k}}_{p,w} + \sum\nolimits_{n \in V_1(p)} \dot{\mu}^{\hat{k}}_{p,w,n} \cdot \big(\dot{v}^{\hat{k}}_n - v_n\big) \ \forall\, p \in \mathcal{P},\, w \in \mathcal{W},\, \hat{k} \in \mathcal{K} && (1) \\
& v_n \geq \lambda^-_n \cdot \bar{v}_n \ \forall\, n \in V_1 && (2) \\
& v_n \leq \lambda^+_n \cdot \bar{v}_n \ \forall\, n \in V_1 && (3) \\
& \theta_{p,w} \in (-\infty, \infty) \ \forall\, p \in \mathcal{P},\, w \in \mathcal{W} && (4) \\
& v_n \in [0, \lambda^+_n] \ \forall\, n \in V_1 && (5) \\
& \bar{v}_n \in \{0, 1\} \ \forall\, n \in V_1 && (6)
\end{aligned}
$$

The objective function in the Benders master problem captures the fixed and variable capital costs to construct additional components in the supply chain as well as the operational costs of all of the Benders subproblems. The first set of constraints in the Benders master problem, Eq. (1), are the valid Benders cuts that are generated in the during k iterations of the procedure, where $\mathcal{K}$ is given by the set $\{1, 2, \ldots, k\}$. It should be highlighted, that due to the addition of the slack commodity sources and sinks in each of the operational scheduling subproblems, there is no need for the addition of feasibility cuts. Equation (2) and (3) bound the nameplate capacities of the components and allow for the inclusion of fixed capital costs. Equation (4) describes the continuous variables that act as an upper bound to the objective values for the Benders subproblems. Equation (5) describes the continuous variables that capture the nameplate capacities of the components. Equation (6) describes the binary variables that indicate whether or not a component is utilized in the energy system or not.

At any point in the iterative Benders procedure, an upper bound can be computed via the following trivial problem.

$$
B\big(\dot{\boldsymbol{v}}, \dot{\boldsymbol{\theta}}\big) = \min_{k \in \mathcal{K}} \Big\{ \sum\nolimits_{n \in V_1 \setminus \mathcal{H}:\ \dot{v}^k_n > 0} \big(\bar{V}_n + V_n \cdot \dot{v}^k_n\big) + \sum\nolimits_{p \in \mathcal{P}} \sum\nolimits_{w \in \mathcal{W}} \dot{\theta}^k_{p,w} \Big\}
$$

3.2 Benders Subproblem – Operational Scheduling Problem

The mathematical programming formulation of the Benders subproblem for a given planning period, $p \in \mathcal{P}$, and scheduling scenario, $w \in \mathcal{W}$, is given by $S_{p,w}(\cdot)$ and is a function of the optimal infrastructure planning decisions, $\tilde{\boldsymbol{v}}$, found in the Benders master problem.

$$S_{p,w}(\tilde{\boldsymbol{v}}) = \min\ J_1 + J_2$$

$$\text{s.t.}\ \sum\nolimits_{a \in \mathcal{A} \cap V_1(p,l,r)} \sigma_{a,r} \cdot x_a^{p,w,\tau_s^-} + \sum\nolimits_{b \in V_1^-(p,l,r)} \gamma_{b,r} \cdot x_b^{p,w,\tau_{s,b}^-} + \ldots \tag{7}$$

$$\sum\nolimits_{c \in \mathcal{C} \cap V_1(p,l,r)} \eta_{c,r}^{p,w,s} \cdot x_c^{p,w,s} + \sum\nolimits_{d \in \mathcal{D} \cap V_1(p,l,r)} \sum\nolimits_{o \in V_2(d)} \phi_{d,o,r}^{p,w,s} \cdot y_{d,o}^{p,w,s} + \ldots$$

$$m_{l,r}^{p,w,s} = \beta_{l,r}^{p,w,s} + n_{l,r}^{p,w,s} + \sum\nolimits_{b \in V_1^+(p,l,r)} x_b^{p,w,s} + \sum\nolimits_{a \in \mathcal{A} \cap V_1(p,l,r)} x_a^{p,w,s} \ \ldots$$

$$\forall\ s \in \mathcal{S},\ l \in \mathcal{L},\ r \in \mathcal{R}(l)$$

$$\sum\nolimits_{\bar{o} \in V_2^-(d,o)} \bar{z}_{d,\bar{o},o}^{p,w,s} = \sum\nolimits_{\bar{o} \in V_2^+(d,o)} \bar{z}_{d,o,\bar{o}}^{p,w,\tau_{s,d,o,\bar{o}}^+} \ \forall\ s \in \mathcal{S},\ d \in \mathcal{D} \cap V_1(p),\ \ldots$$

$$o \in V_2(d) \tag{8}$$

$$\sum\nolimits_{(\bar{o},o) \in E_2(d)} \bar{z}_{d,\bar{o},o}^{p,w,s} \leq 1\ \forall\ s \in \mathcal{S},\ d \in \mathcal{D} \cap V_1(p) \tag{9}$$

$$y_{d,o}^{p,w,s} \geq \zeta_{d,o}^- \cdot \sum\nolimits_{\bar{o} \in V_2^-(d,o)} \bar{z}_{d,\bar{o},o}^{p,w,s}\ \forall\ s \in \mathcal{S},\ d \in \mathcal{D} \cap V_1(p),\ o \in V_2(d) \tag{10}$$

$$y_{d,o}^{p,w,s} \leq \zeta_{d,o}^+ \cdot \sum\nolimits_{\bar{o} \in V_2^-(d,o)} \bar{z}_{d,\bar{o},o}^{p,w,s}\ \forall\ s \in \mathcal{S},\ d \in \mathcal{D} \cap V_1(p),\ o \in V_2(d) \tag{11}$$

$$y_{d,o}^{p,w,s} - y_{d,\bar{o}}^{p,w,\tau_{s,d,\bar{o},o}^-} \leq (\delta_{d,\bar{o},o}^+ - \zeta_{d,o}^+) \cdot \bar{z}_{d,\bar{o},o}^{p,w,s} + \zeta_{d,o}^+ \cdot \bar{v}_d\ \forall\ s \in \mathcal{S},\ \ldots \tag{12}$$

$$d \in \mathcal{D} \cap V_1(p),\ (\bar{o},o) \in E_2(d)$$

$$y_{d,o}^{p,w,s} - y_{d,\bar{o}}^{p,w,\tau_{s,d,\bar{o},o}^-} \geq (\delta_{d,\bar{o},o}^- + \zeta_{d,\bar{o}}^+) \cdot \bar{z}_{d,\bar{o},o}^{p,w,s} - \zeta_{d,\bar{o}}^+ \cdot \bar{v}_d\ \forall\ s \in \mathcal{S},\ \ldots \tag{13}$$

$$d \in \mathcal{D} \cap V_1(p),\ (\bar{o},o) \in E_2(d)$$

$$x_n^{p,w,s} \leq v_n\ \forall\ s \in \mathcal{S},\ n \in V_1(p) \setminus \mathcal{D} \tag{14}$$

$$v_n \geq \lambda_n^- \cdot \bar{v}_n\ \forall\ n \in V_1(p) \tag{15}$$

$$v_n \leq \lambda_n^+ \cdot \bar{v}_n\ \forall\ n \in V_1(p) \tag{16}$$

$$v_n = \tilde{v}_n\ \forall\ n \in V_1(p) \tag{17}$$

$$m_{l,r}^{p,w,s} \in [0,\infty)\ \forall\ s \in \mathcal{S},\ l \in \mathcal{L},\ r \in \mathcal{R}(l) \tag{18}$$

$$n_{l,r}^{p,w,s} \in [0,\infty)\ \forall\ s \in \mathcal{S},\ l \in \mathcal{L},\ r \in \mathcal{R}(l) \tag{19}$$

$$v_n \in [0,\lambda_n^+]\ \forall\ n \in V_1(p) \tag{20}$$

$$x_n^{p,w,s} \in [0,\lambda_n^+]\ \forall\ s \in \mathcal{S},\ n \in V_1(p) \setminus \mathcal{D} \tag{21}$$

$$y_{d,o}^{p,w,s} \in [0,\lambda_d^+]\ \forall\ s \in \mathcal{S},\ d \in \mathcal{D} \cap V_1(p),\ o \in V_2(d) \tag{22}$$

$$\bar{v}_n \in [0,1]\ \forall\ n \in V_1(p) \tag{23}$$

$$\bar{z}_{d,\bar{o},o}^{p,w,s} \in [0,1]\ \forall\ s \in \mathcal{S},\ d \in \mathcal{D} \cap V_1(p),\ (\bar{o},o) \in E_2(d) \tag{24}$$

The objective functions J_1 and J_2 are given by Eq. (25) and Eq. (26) respectively. Equation (25) sums the variable cost to: (i) dispose of excess commodities, raw materials, intermediate products, and or final products, at the the commodity sinks; (ii) purchase additional commodities, raw materials, intermediate products, and or final products, at the the commodity sources; and (iii) operate the non-mode based processes, storage units, and the transportation mechanisms. Equation (26) sums the fixed and variable cost to operate the mode based processes.

$$\begin{aligned} J_1 = &\sum\nolimits_{s\in\mathcal{S}}\sum\nolimits_{l\in\mathcal{L}}\sum\nolimits_{r\in\mathcal{R}(l)} \Delta_{l,r}^{p,w,s}\cdot m_{l,r}^{p,w,s} + \ldots \\ &\sum\nolimits_{s\in\mathcal{S}}\sum\nolimits_{l\in\mathcal{L}}\sum\nolimits_{r\in\mathcal{R}(l)} P_{l,r}^{p,w,s}\cdot n_{l,r}^{p,w,s} + \ldots \\ &\sum\nolimits_{s\in\mathcal{S}}\sum\nolimits_{n\in V_1(p)\setminus\mathcal{D}} X_{n}^{p,w,s}\cdot x_{n}^{p,w,s} + \ldots \end{aligned} \tag{25}$$

$$\begin{aligned} J_2 = &\sum\nolimits_{s\in\mathcal{S}}\sum\nolimits_{d\in\mathcal{D}\cap V_1(p)}\sum\nolimits_{o\in V_2(d)} Y_{d,o}^{p,w,s}\cdot y_{d,o}^{p,w,s} + \ldots \\ &\sum\nolimits_{s\in\mathcal{S}}\sum\nolimits_{d\in\mathcal{D}\cap V_1(p)}\sum\nolimits_{(\bar{o},o)\in E_2(d)} \bar{Z}_{d,\bar{o},o}^{p,w,s}\cdot \bar{z}_{d,\bar{o},o}^{p,w,s} \end{aligned} \tag{26}$$

The first set of constraints, Eq. (7), are analogous to the material balance constraints, Eq. (S1), in the fullspace formulation, P, that is given in the Supplementary Material section. The second set of constraints, Eqs. (10), (11), (12), (13), and (14) correspond to the operational bounding and ramping constraints, Eqs. (S4), (S5), (S6), (S7), and (S8). The third set of constraints, Eq. (15) and Eq. (16), correspond to the indicator constraints, Eq. (S9) and Eq. (S10). The final constraint, Eq. (17), fixes the name plate capacity of the components to the value passed into the operational scheduling subproblem. Similarly, Eqs. (18) - (24) correspond to Eqs. (S8), (S11) - (S17).

4 Valid Inequalities

When the fullspace problem is decomposed, the superstructure of the energy system and the interdependencies between processes are not explicitly captured in the Benders master problem. The Benders master problem only becomes implicitly aware of them, through the addition of the Benders cuts. This in turn, can cause a delay in the time it takes for the procedure to fully converge. Consequently, using the following mild assumption, we have developed a set of valid inequalities that can be applied a priori to the Benders master problem.

Assumption 1 *The central planner on average, for each scheduling scenario, would rather produce a material than purchase it from a slack commodity source – if the commodity under consideration has an associated strictly positive objective coefficient for its corresponding slack source.*

For each component that does not act as a storage unit, $n \in V_1 \setminus \mathcal{A}$, an auxiliary continuous positive variable, $u_n^{p,w}$, is created that acts as the average set point of the component, $n \in V_1 \setminus \mathcal{A}$, during the scheduling scenario, $w \in \mathcal{W}$, for a given planning period, $p \in \mathcal{P}$.

$$u_n^{p,w} \in [0, \lambda_n^+] \;\forall\; p \in \mathcal{P},\; w \in \mathcal{W},\; n \in V_1(p) \setminus \mathcal{A} \tag{27}$$

Equation (28) is an aggregated material balance constraint and captures the average material consumption, production, and transportation dynamics for each commodity in each planning period, scheduling scenario, and location – were $\mathcal{R}^{+}(l)$ is the subset of resources in a location that have a strictly positive associated slack cost.

$$\sum\nolimits_{b\in V_1^{-}(p,l,r)} \gamma_{n,r}\cdot u_b^{p,w} + \sum\nolimits_{c\in V_1(p,l,r)\cup\mathcal{C}} \acute{\eta}_{c,r}^{p,w}\cdot u_c^{p,w} + \sum\nolimits_{d\in V_1(p,l,r)\cup\mathcal{D}} \acute{\phi}_{d,r}^{p,w}\cdot u_d^{p,w} \geq \ldots \tag{28}$$
$$|\mathcal{S}|^{-1}\cdot\sum\nolimits_{s\in\mathcal{S}} \beta_{l,r}^{p,w,s} + \sum\nolimits_{b\in V_1^{+}(p,l,r)} u_b^{p,w}\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ l\in\mathcal{L},\ r\in\mathcal{R}^{+}(l)$$

The aggregated parameters, $\acute{\eta}_{c,r}^{p,w}$ and $\acute{\phi}_{d,r}^{p,w}$ utilized in the Eq. (28), where generated to ensure that if a process is producing a material its maximum efficiency factor for the scheduling horizon and if a process is consuming a material its minimum efficiency factor for the scheduling horizon is utilized in the aggregated material balance constraint.

$$\acute{\eta}_{c,r}^{p,w} = \begin{cases} \max_{s\in\mathcal{S}}\left\{\eta_{c,r}^{p,w,s}\right\} & \text{if } c\in V_1^{+}(r) \\ \max_{s\in\mathcal{S}}\left\{\eta_{c,r}^{p,w,s}\mid \eta_{c,r}^{p,w,s}<0\right\} & \text{if } c\in V_1^{-}(r) \end{cases}$$

$$\acute{\phi}_{d,r}^{p,w} = \begin{cases} \max_{s\in\mathcal{S},\ o\in V_2(d)}\left\{\phi_{d,o,r}^{p,w,s}\right\} & \text{if } c\in V_1^{+}(r) \\ \max_{s\in\mathcal{S},\ o\in V_2(d)}\left\{\phi_{d,o,r}^{p,w,s}\mid \phi_{d,o,r}^{p,w,s}<0\right\} & \text{if } c\in V_1^{-}(r) \end{cases}$$

The valid inequality given in Eq. (29) integrates these new axillary variables with the infrastructure planning variables.

$$u_n^{p,w} \leq v_n\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ n\in V_1(p)\setminus\mathcal{A} \tag{29}$$

Proof 1 *Consider Assumption 1 and observe that: (i) firstly, if the material balance constraints for the fullspace formulation, given by Eq.* (S1)*, that map to the same planning period, $p\in\mathcal{P}$, scheduling scenario, $w\in\mathcal{W}$, location, $l\in\mathcal{L}$, and material, $r\in\mathcal{R}^{+}(l)$, are aggregated into a single set of constraints; (ii) secondly, if the equality signs in these new constraints are replaced by the greater than or equal to signs; (iii) thirdly, if the scheduling period dependent conversion factors for the non-mode based and mode based components are replaced by their respective aggregated best case conversion factors in these new constraints; (iv) fourthly, if the scheduling dependent variables that track their respective sets points are replaced with the new axillary variable, $u_n^{p,w}$, in these new constraints; (v) fifthly, if the variables, $m_{l,r}^{p,w,s}$ and $n_{l,r}^{p,w,s}$, that track the amount of material purchased and disposed at the slack source and slack sinks respectively are dropped in accordance with Assumption 1 in these new constraints; and (vi) finally, if the total amount of material demanded in the location is divided by the cardinality of the scheduling horizon in these new constraints, then the these new constraints, correspond to the constraints given by Eq.* (28)*, and in conjunction with Eq.* (27) *and Eq.* (29) *serve as a valid relaxation the fullspace formulation.*

5 Machine Learning Framework

In this section, the machine learning framework utilized to increase the computational performance of the Benders decomposition procedure is presented. The basic premise behind the machine learning framework is that in each iteration of the Benders decomposition procedure not all of the subproblems need to be solved – the only ones that need to be solved are the ones that have not converged to their optimal value. This is especially true when there are more subproblems than threads on the machine that is solving the optimization problem due to the fact that subproblems that have already converged are being resolved unnecessarily. Therefore, in each iteration of the Benders decomposition procedure, the goal of the machine learning framework is to first identify the subproblems that have to yet to fully converge at the current iteration underway and then to select a subset of these subproblems to be solved in the current iteration based upon the historical variance of their respective objective function evaluations – it should be highlighted, that the convergence proof, proving that if the objective function evaluations of each subproblem in two sequential iterations of the Benders procedure is identical convergence has been reached is omitted and interested readers can see [7].

The combined Benders decomposition procedure and machine learning procedure is illustrated in the Figure 1. The framework is a function of four inputs: (i) the convergence tolerance of the Benders decomposition procedure, `convergeTol`; (ii) the maximum number of historical objective evaluations utilized to compute the variance within the subproblems objective function, `lenOfVar`; (iii) the number of candidate subproblems that should be generated in each iteration of the Benders decomposition procedure, `numOfCanidates`; and (iv) a parameter that controls the frequency in which all the subproblems are solved, which in turn allows a feasible upper bound to be computed, `modBase`.

In the preprocessing stage of the the framework, the valid inequalities, given in Eqs (27) - (29), are inserted directly into the Benders master problem, $B_k(\cdot, \cdot, \cdot)$. Subsequently, the local parameters that are utilized in Benders decomposition procedure are initialized, where: (i) k is a iteration counter; (ii) $\boldsymbol{v}_k$ is the optimal decisions to the Benders master problem in the k iteration; (iii) $\boldsymbol{\theta}_k$ is the objective value evaluations of the Benders subproblems in the k iteration; and (iv) $\boldsymbol{\mu}_k$ is the optimal dual variables found in the Benders subproblems in the k iteration.

In the first phase of the Benders decomposition procedure, the Benders master problem is solved using the historical: (i) optimal decisions to the Benders master problem; (ii) objective value evaluations of the Benders subproblems; and (iii) optimal dual variables found in the Benders subproblems. Once the Benders master problem is solved, the framework transitions into the machine learning portion of the framework to determine which of the Benders subproblems should be solved. If it first iteration of the procedure or if $k \,\texttt{mod}\, \texttt{modBase}$ is equal to zero, all of the Benders current iteration; otherwise the machine learning framework is utilized to determine which of the Benders subproblems should be solved based upon historical variance of the objective function evaluations of the Benders subproblems. Then manner in which machine learning framework framework identifies these subproblems is given in Algorithm 1.

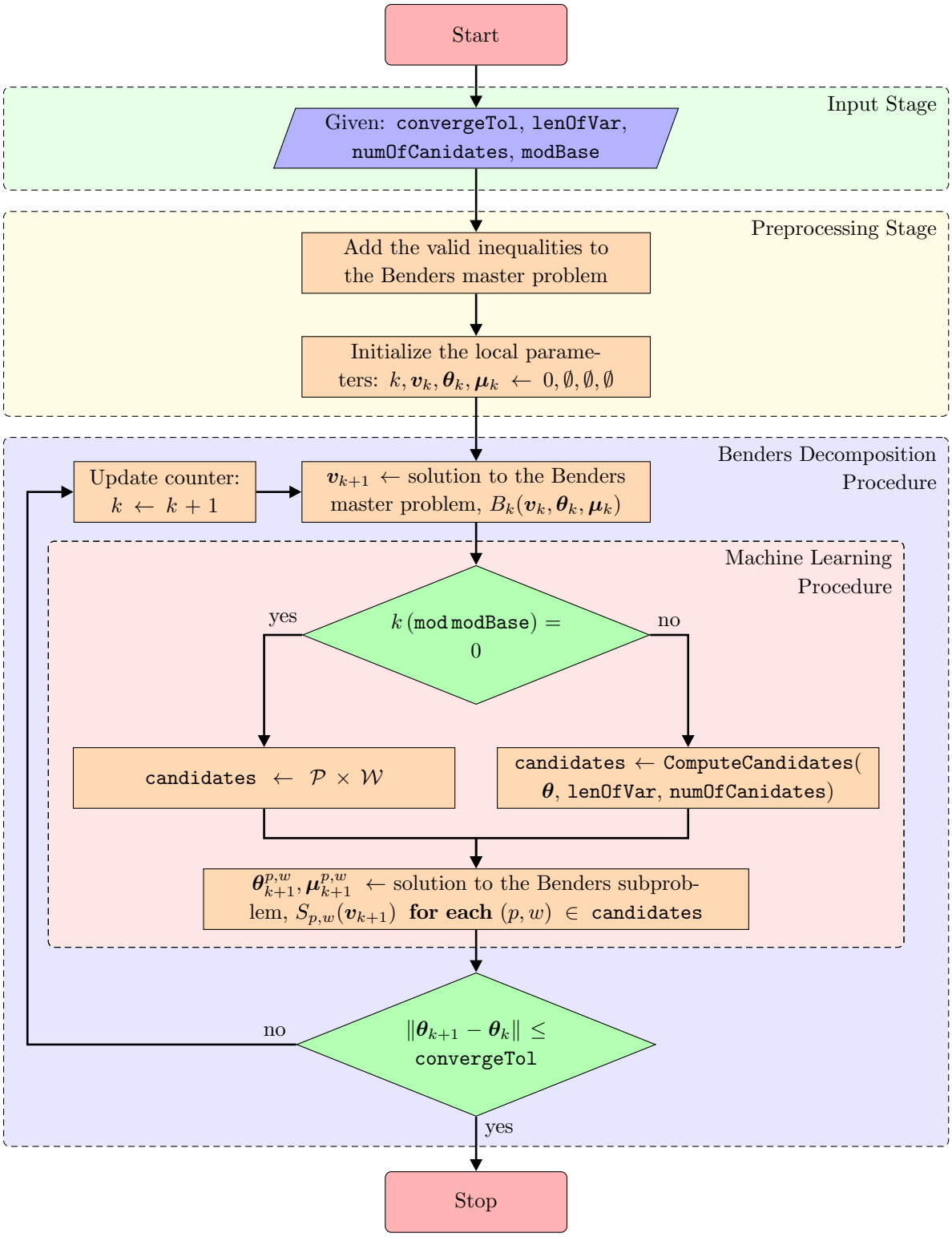

Figure 1: Benders decomposition procedure with the machine learning framework.

Algorithm 1: Machine learning procedure utilized to compute candidate subproblems

input : $\boldsymbol{\theta}$ – historical objective value evaluations of the subproblems
`lenOfVar` – maximum number of objective value evaluations considered
`numOfCanidates` – number of candidate subproblems generated
output: `candidateSubproblems` – candidate subproblems

1 **Function** `ComputeCandidates(`$\boldsymbol{\theta}$`, lenOfVar, numOfCanidates):`
2 $\quad\boldsymbol{\omega} \leftarrow \emptyset$
3 $\quad$**foreach** $(p, w) \in \mathcal{P} \times \mathcal{W}$ **do**
4 $\quad\quad\boldsymbol{\omega}_{p,w} \leftarrow$ `Variance`$(\{\boldsymbol{\theta}_k^{p,w} \mid k \in \{(\texttt{Length}(\boldsymbol{\theta}^{p,w}) - \texttt{lenOfVar}) \ldots \texttt{Length}(\boldsymbol{\theta}^{p,w})\}\})$
5 $\quad$`candidateSubproblems` $\leftarrow$ `Sample`$(\mathcal{P} \times \mathcal{W},$ `numOfCanidates`$, \boldsymbol{\omega})$
6 $\quad$**return** `candidateSubproblems`

Algorithm 1, is a function of: (i) the historical objective value evaluations of the subproblems, $\boldsymbol{\theta}$; (ii) the maximum number of historical objective evaluations utilized to compute the variance within the subproblems objective function, `lenOfVar`; and (iii) the number of candidate subproblems that should be generated in each iteration of the Benders decomposition procedure, `numOfCanidates`. In the first phase of the procedure, as seen in line 2, an empty data structure is created to store the variance for each of the subproblems. In the second phase of the procedure, as seen in lines 3 and 4, the variances for each of the subproblems are computed. And in the final phase of the procedure, as seen in line 5, a subset of the candidate subproblems with a cardinality of `numOfCanidates` is randomly computed with weights of each subproblem being selected being based upon their historical variance, via the `Sample`$(\cdot, \cdot, \cdot)$ function, where `Sample`$(\cdot, \cdot, \cdot)$ is function that accepts three inputs: (i) the first input is a list of items that can be sampled without replacement; (ii) the second input is the number of items that will be sampled; and (iii) the third input is a data structure that corresponds to the probability that an item in the input list will be selected.

Once the candidate subproblems have been identified, they can be solved in serial or parallel depending on the algorithm is implemented. After the candidate subproblems have been solved, a test is conducted to determine if the algorithm converged successfully or not by examining the objective function evaluations of each subproblem in their last two sequential iterations and determining if there difference is less than the convergence tolerance. If the difference is less than the convergence tolerance, the framework is complete; however, if that is not the case, the Benders decomposition procedure begins another iteration.

6 Case Study

In this section, we present a case study to illustrate the effectiveness of the proposed solution strategy. Consider a central planner who is trying to develop an energy system that spans multiple locations, Beaumont, TX, Corpus Christi, TX, Long Beach, CA, and Toms River, NJ, over the course of multiple planning periods, 2025 to 2030, 2030 to 2035, 2035 to 2040, and 2040 to 2045. The central planner would like to develop an energy system that produces electricity, via solar and wind farms, for electrical utility markets. The electricity can also be utlized to provide power to chemical plants that produce H-DECs that can act as fuels for transportation markets. A location

agnostic process flow-sheet for the energy system is given in Figure 2. In the flow-sheet triangular arrow heads indicate that material can flow into the component, square arrow heads indicate that material can either flow into or out of the component, and no arrow heads indicate that material can flow out of the component.

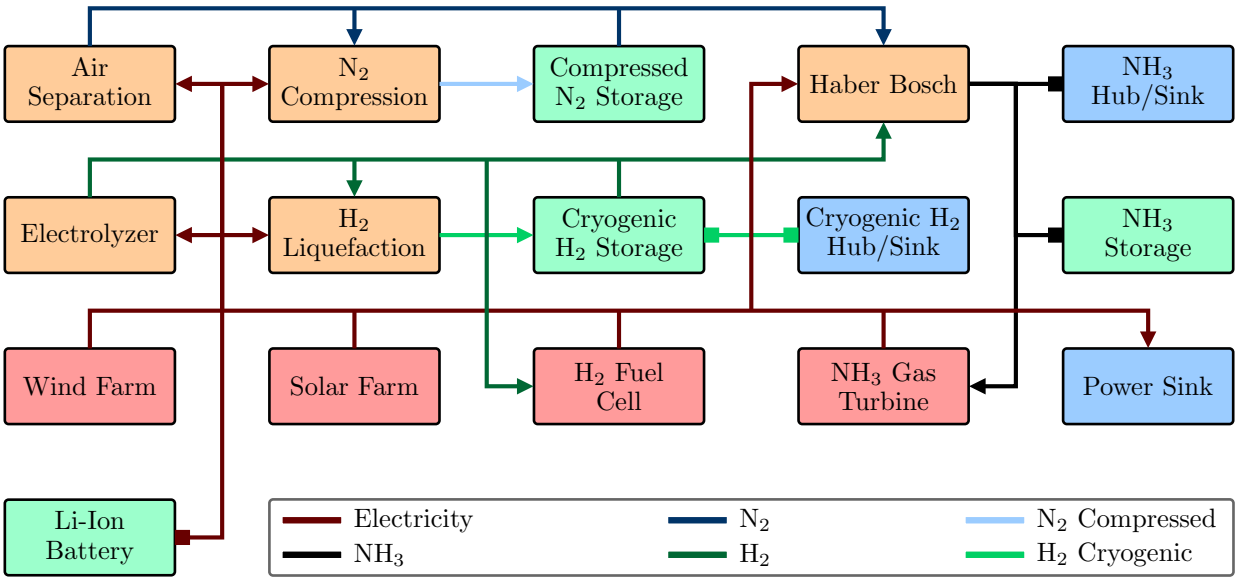

Figure 2: Generic superstructure of the energy system.

The central planner would like to produce ammonia and hydrogen in Beaumont and or in Corpus Christi to meet the hydrogen based fuel demands in Long Beach and Toms River. The central planner also has green electricity demands that must be satisfied in Long Beach and Toms River. To meet these demands the central planner would like to harness the high solar and wind potentials along the Gulf Coast of Texas to possibly produce additional H-DECs that can be transported to Long Beach and Toms River and converted to electricity to meet help meet the respective locations electricity demands. In the event that it is not economical to produce H-DECs in Texas and utilize them as feed stocks for electricity generators in Long Beach and Toms River, the central planner would like to have the option to produce additional green electricity locally in Long Beach and Toms River via solar and or wind farms and to store it if necessary, in Lithium-Ion battery banks. The discrete nameplate capacities for the air separation units and the Haber-Bosch processes that the central planner has elected to choose from is given by 20.3 [MT/hr], 23.6 [MT/hr], and 27.0 [MT/hr] and 15 [MT/hr], 17.5 [MT/hr], and 20 [MT/hr] respectively. The nominal demands for ammonia, electricity, and hydrogen for Long Beach and Toms River over the 4 planning periods is given in Table 1.

The conversions factors, cost functions, and ramping limitations for the components were taken from an earlier case study in the literature, Allen et al. [3]. The hourly capacity factors for the solar and wind farms where taken from the SIND Toolkit[13] and WIND toolkit[10] respectively. The scheduling scenarios where generated from the temporal dependent capacity factors and material

Table 1: Chemical and Energy Demands in Long Beach and Toms River

		Planning Period			
Location	Material	2025 – 2030	2030 – 2035	2035 – 2040	2040 – 2045
Long Beach, CA	Electricity [GW]	1.00	1.17	1.33	1.5
	H_2 [MT/hr]	0.10	0.22	0.46	1.00
	NH_3 [MT/hr]	10.0	13.3	16.7	20.0
Toms River, NJ	Electricity [GW]	1.00	1.17	1.33	1.50
	H_2 [MT/hr]	0.00	0.00	0.00	0.00
	NH_3 [MT/hr]	5.00	8.33	11.7	15.0

demands via k-means clustering using the Scikit-learn toolbox [23]. We refer the reader to an earlier work, Allen et al. [3], which describes in detail how the scheduling scenarios, which will henceforth be referred to as representative days.

7 Computational Experiments

In this section, we illustrate the effectiveness of the proposed valid inequalities and the machine learning framework in the context of the aforementioned case study. Specifically, we conduct a set of 6 computational experiments wherein we solve the aforementioned case study: (i) in the first set of experiments, we solve the fullspace MILP formulation, given by P, which can be seen in the supplementary materials section, with integrality relaxed in the operational scheduling problems; (ii) in the second set of experiments, we solve the fullspace MILP formulation, given by P, with the a priori addition of the valid inequalities and with integrality relaxed in the operational scheduling problems; (iii) in the third set of experiments, we examine the performance of the multi-cut Benders decomposition procedure; (iv) in the forth set of experiments, we examine the performance of the multi-cut Benders decomposition procedure with the a priori addition of the valid inequalities; (v) in the fifth set of experiments, we examine the performance of the multi-cut Benders decomposition procedure with integrated machine learning framework, which selects the subproblems to be solved in each iteration; and (vi) in the sixth and final set of experiments we examine the performance of the multi-cut Benders decomposition procedure with he a priori addition of the valid inequalities and integrated machine learning framework.

In each solution procedure we parameterize the number of scheduling scenarios, that act as representative days each with 24 hourly time periods, embedded within the optimization problem to be between 1 and 10, which in turn gives rise to a total of 60 different test instances – 10 for each solution procedure. For each of the solution procedures the convergence tolerance, `convergenceTol` was set to 1.0 [%] and the maximum time limit, `timeLimit`, was set for a 1000 [seconds], where the formula for to comupute the convergence tolerance is given by: (`upperBound` − `lowerBound`)/`upperBound`. Moreover, for the solution procedures that utilize the machine learning framework, we have set: (i) the maximum number of objective value evaluations considered, `lenOfVar`, to be equal to 5; (ii) the number of candidate subproblems generated in each

iteration, `numOfCanidates`, to be equal to 10; and (iii) the divisor utilized to control the frequency of computing upper bounds, `modBase`, to be equal to 10.

Table 2 summarizes the number of linear algebraic constraints, binary variables, and continuous variables in the fullspace mathematical programming model as a function of the number of scheduling scenarios. It should be noted, that the number of linear algebraic constraints, binary variables, and continuous variables are all in thousands. For instance, when the fullspace problem is solved using 2 scheduling scenarios there are approximately 166,200 linear algebraic constraints, 27,000 binary variables, and 50,700 continuous variables.

Table 2: Summary statistics describing the size of the test instances in thousands.

	Number of Representative Scheduling Scenarios [Days]									
Type	1	2	3	4	5	6	7	8	9	10
Constraints	83.3	166.2	249.2	332.1	415.1	498.0	580.9	663.9	746.8	829.8
Binary variables	13.6	27.0	40.5	53.9	67.4	80.8	94.2	107.7	121.1	134.6
Continuous variables	25.4	50.7	75.9	101.2	126.4	151.7	176.9	202.2	227.4	252.6

The mathematical programming models were constructed utilizing the JuMP toolbox in Julia and were solved utilizing Gurobi version 9.5 [11, 26]. The experiments were performed on a machine with an Intel i7-11850H Processor, 32 GBs of RAM, and a maximum number of available processors of 10.

7.1 Performance of the Solution Procedures

In this subsection, the results to the 6 sets of computational experiments are presented. Figure 3 is a conglomeration of 6 different sub-figures, where each sub-figure is utilized to illustrate how the run time in seconds and the number of representative scheduling scenarios in days affects the MIP gap [%] of the respective set of computational experiments. In each of these sub-figures, the color of the bar represents the MIP gap [%] as a function of the number of representative scheduling scenarios and the run time. It should be noted, that the bar is colored black, when: (i) either the lower bound to problem has yet to be calculated; (ii) the upper bound has yet to be calculated; or (iii) the MIP gap [%] between the two is greater than 100 [%].

Figure 3.a, illustrates the computational performance of Gurobi 9.5 when it solves the fullspace formulation with integrality relaxed in the operational scheduling subproblems. From inspection of this sub-figure, it is clear that the "state-of-the-art" MIP solver is only able to close the MIP gap [%] to a value less than 100 [%] for the 2 smallest test instances and it is only able to solve the smallest of the 2 to the desired optimality gap of 1.0 [%].

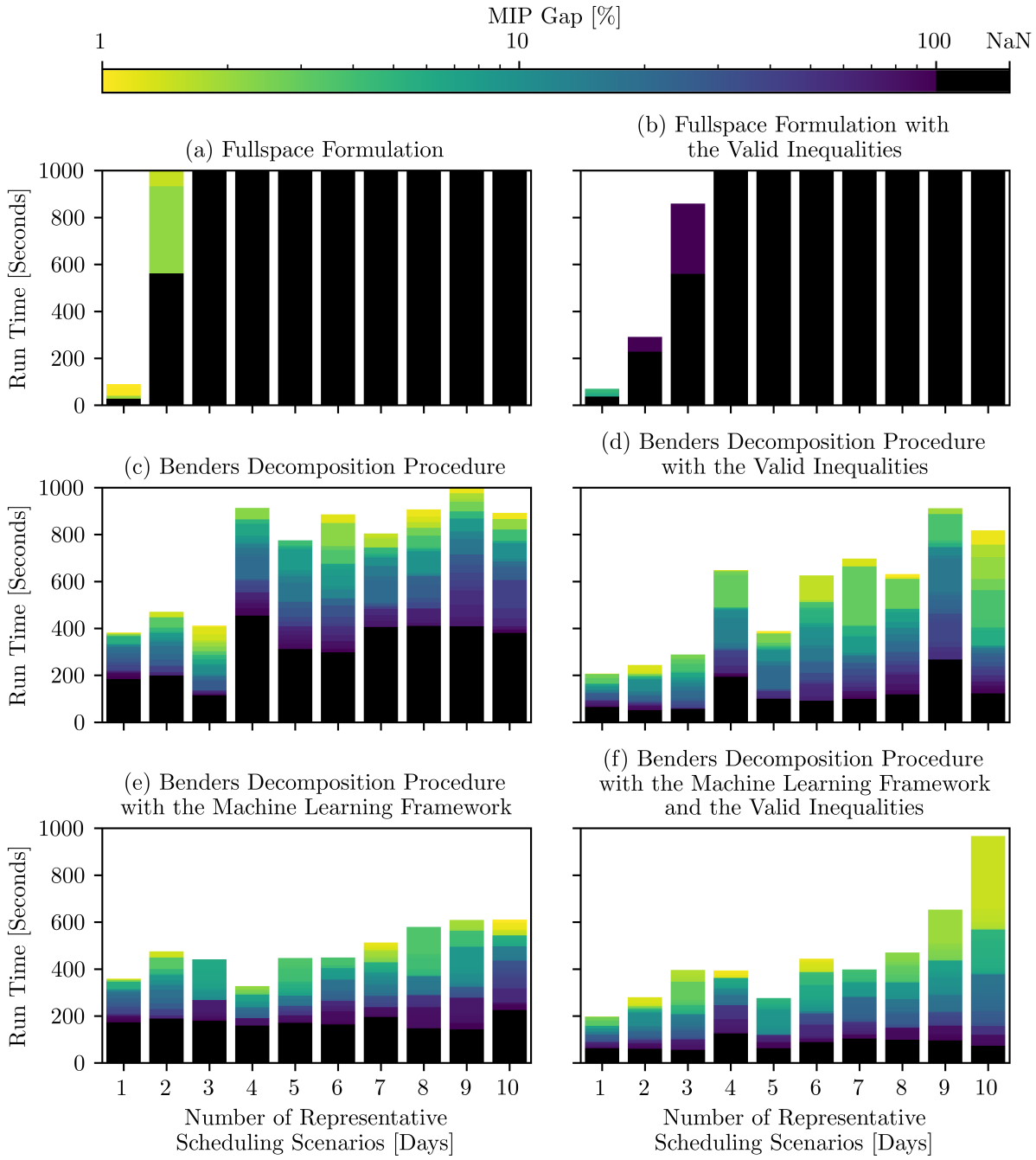

Figure 3: Convergence plots for the 6 sets of computational experiments.

Figure 3.b illustrates the computational performance of "state-of-the-art" MIP solver when it solves the fullspace formulation with integrality relaxed in the operational scheduling subproblems and the inclusion of the valid inequalities to the formulation. From inspection of this sub-figure, it is evident that the valid inequalities slightly improve the computational performance the MIP solver. Specifically the inclusion of the valid inequalities allow for solutions of 3 of the 10 test instances to be found that have an optimality gap of less than 1 [%] within the desired time limit

of 1000 [seconds].

Figure 3.c illustrates the computational performance of the Benders decomposition procedure. It is evident from this sub-figure that the Benders decomposition procedure is able to close the MIP gap to less than 1 [%] for 9 out of the 10 test instances within the desired time limit of 1000 [seconds].

Figure 3.d illustrates the computational performance of the Benders decomposition procedure with the inclusion of the valid inequalities applied to the Benders master problem. From inspection of this sub-figure, it is clear that the addition of the valid inequalities allow for all of the 10 test instances for be solved to optimality. Furthermore, each test instance converges faster when the Benders decomposition procedure includes the valid inequalities as opposed to when they are excluded.

Figure 3.e illustrates the computational performance of the Benders decomposition procedure with the inclusion of the machine learning framework. In all but two of the test instances, the inclusion of the machine learning framework allows the procedure to converge faster than it would have otherwise. Moreover, for the test instances with 6 or more representative scheduling scenarios, the machine learning framework outperforms the the procedure when just of the valid inequalities are utilized

Table 3: Percent decrease in run time for a given solution procedure to reach a specified MIP gap [%] for a given number of scheduling when compared to the Benders decomposition procedure.

Solution procedure	MIP Gap [%]	Number of Representative Scheduling Scenarios [Days]									
		1	2	3	4	5	6	7	8	9	10
Benders	1	45.7	48.1	29.9	29.0	49.9	29.4	13.3	30.4	8.8	8.4
decomposition	5	54.0	49.5	24.3	42.2	57.6	35.7	44.5	34.6	12.1	49.3
procedure with the	10	57.9	54.7	20.0	39.3	56.9	34.0	57.0	36.6	8.3	54.0
valid inequalities	25	58.8	62.2	33.9	48.0	72.0	57.8	52.5	44.7	32.4	56.8
	50	61.1	62.2	53.6	55.6	70.1	60.1	65.1	55.8	31.6	54.8
	100	64.7	74.2	50.2	57.4	67.8	69.1	75.4	71.0	34.6	67.8
Benders	1	5.9	-0.9	-7.1	64.1	42.4	49.3	36.2	36.1	39.1	31.6
decomposition	5	5.8	1.4	-57.9	65.5	54.0	40.1	40.3	49.3	42.9	29.6
procedure with the	10	5.7	2.9	-29.8	67.8	54.7	37.3	46.3	41.7	53.9	30.0
machine learning	25	5.7	4.9	-91.3	66.9	52.2	44.5	43.4	46.7	54.7	30.2
framework	50	5.8	5.1	-100.8	64.9	49.8	35.3	50.8	44.6	40.7	28.4
	100	5.7	5.1	-56.6	65.1	45.3	44.8	51.7	64.0	65.0	40.8
Benders	1	48.2	40.5	3.9	56.8	64.4	49.9	50.5	48.2	34.7	-8.4
decomposition	5	55.8	40.6	18.8	57.2	64.5	50.9	52.1	53.6	49.7	26.5
procedure with the	10	59.5	46.7	5.9	58.5	76.7	60.8	59.2	56.5	59.5	46.9
machine learning	25	60.3	55.0	-0.9	59.2	76.6	57.7	62.3	61.3	69.9	64.6
framework and the	50	62.7	55.0	24.9	60.7	72.7	60.1	63.9	69.2	66.1	64.4
valid inequalities	100	66.0	69.7	51.5	72.4	80.0	70.3	74.4	76.0	76.6	80.8

Figure3.f illustrates the computational performance of the Benders decomposition procedure

with the inclusion of the machine learning framework and the addition of the valid inequalities. From inspection of this sub-figure, the by incorporating both the machine and the valid inequalities allow dramatically increases the computational performance of the Benders decomposition procedure.

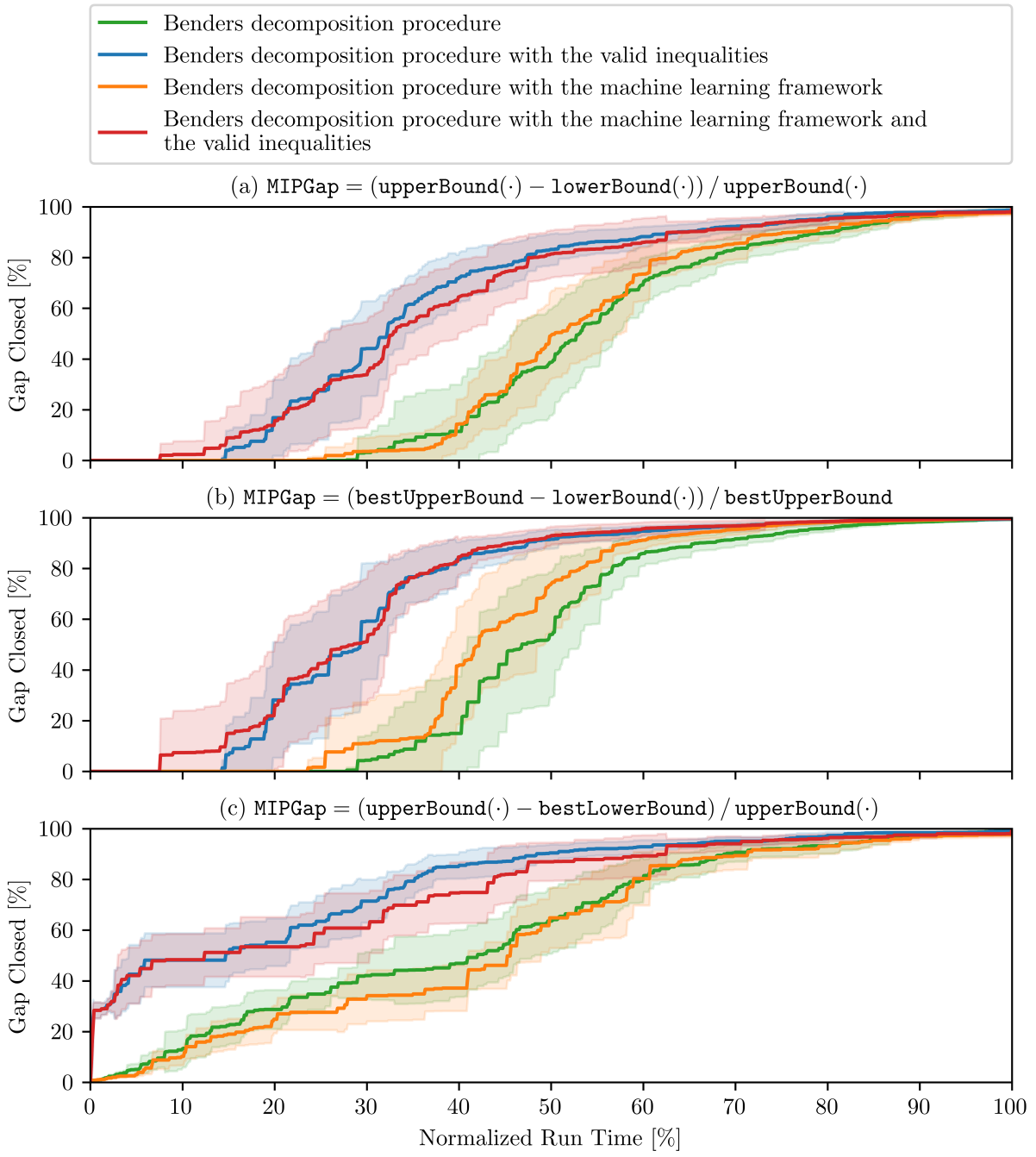

Figure 4: Normalized convergence plots with 95 [%] confidence intervals for the 4 sets of solution procedures that utilize Benders decomposition.

Table 3 illustrates how including the valid inequalities to the Benders master problem, incorporating the machine learning framework, or some combination thereof increases the computational performance of the Benders decomposition procedure, when compared to the Benders decomposition procedure without these additional features. From inspection of the table, it is clear than the inclusion of the valid inequalities allow an initial solution feasible solution to be found on average approximately 63 [%] times faster and an optimal solution with a MIP gap of less 1 [%] can be found approximately 29 [%] times faster. When the machine learning framework is applied to the test instances with 6 or more representative scheduling scenarios a solution with a MIP gap of less 1 [%] can be found approximately 38 [%] times faster. Moreover, when both the valid inequalities and the machine learning framework is integrated into the Benders decomposition procedure an initial solution with a MIP gap of 100 [%] can be found approximately 72 [%] times faster and a solution with a MIP gap of 5 [%] can be found approximately 47 [%] faster.

Figure 4 illustrates how the 4 solution frameworks that employ the Benders decomposition procedure converge utilizing different methods to compute their respective MIP gap. To ensure an apples to apples comparison, the run times for each set of test instances for each of the 4 solution frameworks have been normalized to their respective time that they take to reach an optimality gap of less than 1 [%]. Then for each of the solution procedures, the mean percentage of the MIP gap closed of the 10 test instances is plotted along with its respective confidence interval.

In Fig. 4.a the MIP gap for each of the 4 solution frameworks is computed by utilizing the standard equation, $(\texttt{upperBound}(\cdot) - \texttt{lowerBound}(\cdot)) \,/\, \texttt{upperBound}(\cdot)$, where $\texttt{upperBound}(\cdot)$ is the best feasible solution found at the time under consideration and $\texttt{lowerBound}(\cdot)$ is the tightest relaxed solution found at the time under consideration. From inspection of this sub-figure, it is clear that the valid inequalities effect the closure of the MIP gap more so than the machine learning framework.

In Fig. 4.b the MIP gap for each of the 4 solution frameworks is computed by the augmenting the standard equation, $(\texttt{bestUpperBound} - \texttt{lowerBound}(\cdot)) \,/\, \texttt{bestUpperBound}$, by replacing $\texttt{upperBound}(\cdot)$ with `bestUpperBound`, where `bestUpperBound` is the best known upper bound to the problem. This in turn, allows for a clearer idea of how the lower bounds for each of the 4 solution frameworks converge to the optimal solution, given by `bestUpperBound`. From inspection of this sub-figure, it is clear that when the valid inequalities are included, the machine learning framework is able to slightly decrease the time that it takes to converge to the optimal solution.

In Fig. 4.c the MIP gap for each of the 4 solution frameworks is computed by the augmenting the standard equation, $(\texttt{upperBound}(\cdot) - \texttt{lowerBound}(\cdot)) \,/\, \texttt{upperBound}(\cdot)$, by replacing $\texttt{lowerBound}(\cdot)$ with `bestLowerBound`, where `bestLowerBound` is the best known lower bound to the problem. By altering the equation in this manner, allows for a clearer picture as to how the upper bounds for each of the 4 solution frameworks converge to the optimal solution. From inspection of this sub-figure, it is evident that the valid inequalities allows for the MIP Gap to be closed by approximately 30 [%] almost immediately.

7.2 Convergence of the Machine Learning Framework

In this subsection, the manner in which the subproblems converge utilizing the machine learning framework is presented. Figure 5 illustrates how each of the subproblems converge for the test instance utilizing 4 scheduling scenarios.

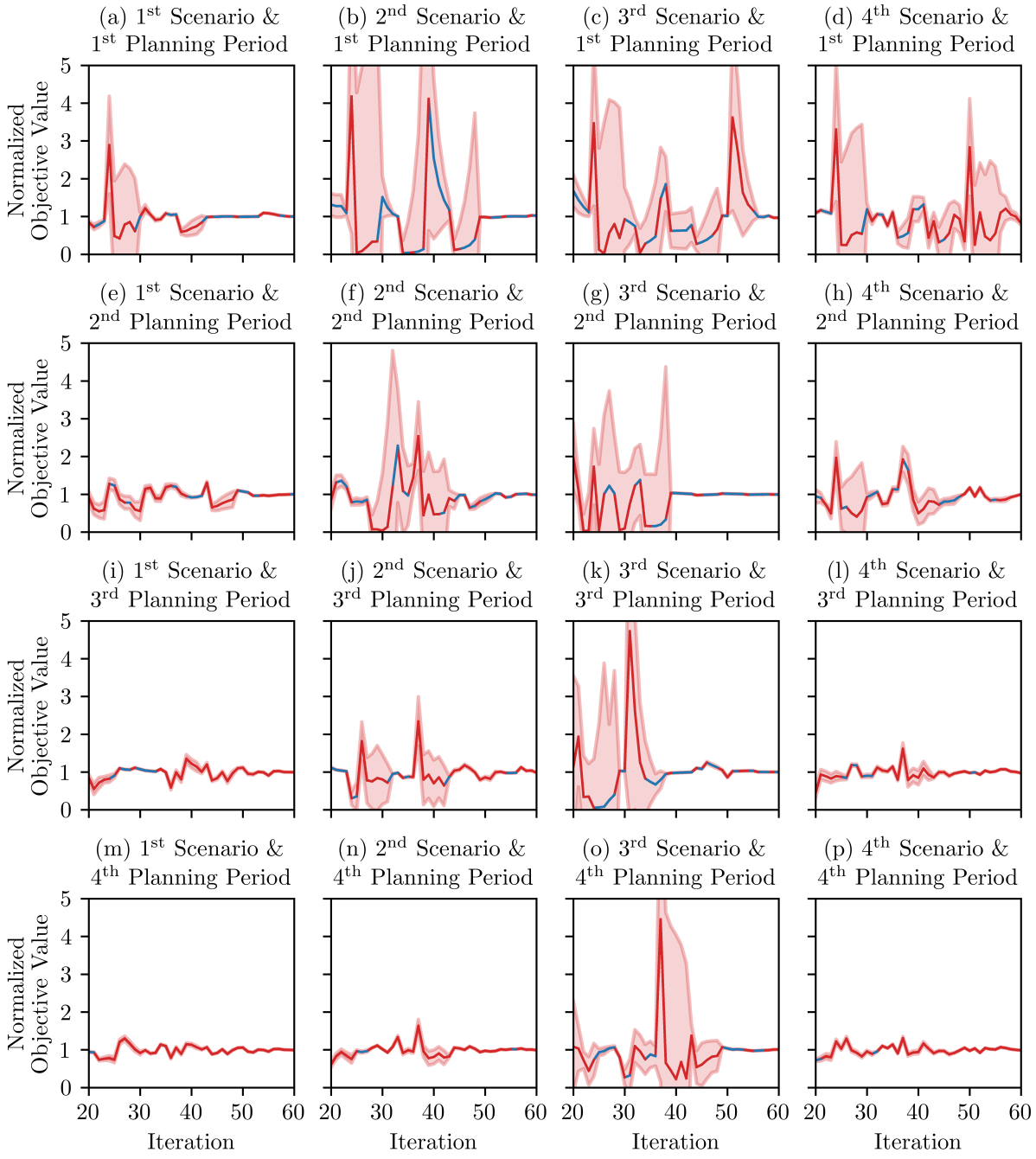

Figure 5: Convergence of the Benders subproblems using the machine learning framework.

In each of the sub-figures in Fig. 5 the x-axis represents the iteration in the Benders decomposi-

tion procedure and the y-axis represents the normalized objective value evaluation of the subproblem under consideration. The objective value was normalized by utilizing by dividing the current objective value by its rolling maximum value. The color of the line indicates whether or not the subproblem was resolved during the respective iteration of the Benders decomposition procedure. For instance, in Fig. 5.i, which represents the subproblem for the 1^{st} scheduling scenario of the 3^{rd} planning period, the scheduling subproblem was solved in the last iteration of the Benders decomposition procedure, while in Fig. 5.b,, which represents the subproblem for the 2^{nd} scheduling scenario of the 1^{st} planning period, the scheduling subproblem was ignored and subsequently not solved in the last iteration of the Benders decomposition procedure. The is clear from the fact that the line is red at the 60^{th} iteration in the former case and bestLowerBound at the 60^{th} iteration in the later case. The width of the upper and lower filled regions surrounding the line indicating the normalized normalized objective value evaluation signifies the rolling 95 [%] confidence interval. It should be noted that the number of historical iterations utilized in computing the rolling maximum value and the rolling 95 [%] confidence interval was set to 5, which corresponds to `lenOfVar` in Fig. 1 and Algorithm 1.

8 Concluding Remarks

In this work, we present a solution framework for developing large-scale energy systems that can be expanded over the course of multiple planning periods. Traditionally, these problems are posed as integrated infrastructure planning and operational scheduling problems and solved via MIP solvers. Unfortunately, this class of problems is extremely computational difficult to solve even with a "state-of-the-art" MIP solver. Therefore, many practitioners utilize decomposition strategies such as the Benders decomposition. However, when these problems are decomposed they are still computationally difficult to solve.

Consequently, we have derived a set of valid inequalities that can be applied a priori to the Benders master problem, and we have developed a machine learning framework to identify which Benders subproblems should be solved in each iteration of the Benders decomposition procedure. To illustrate the effectiveness of the two different frameworks we have generated a case study and ran a set of computational experiments. The case study centers around a green energy system that produces hydrogen based fuels and H-DECs for transportation markets and utilities markets respectively in regions with high renewable energy potentials and transports them to regions with low renewable energy potentials or regions that contain other characteristics that make generating green fuels and green electricity locally uneconomical.

The results show that the valid inequalities significantly strengthens the initial bounds to the problem and that the machine learning framework is able to identify the subproblems that should be solved more aggressively, while simultaneously identifying the subproblems whose solutions marginally affect and or improve the convergence of the Benders decomposition procedure. It should be noted, that the valid inequalities and the machine learning framework can be applied to general supply chain problems that have been decomposed by Benders decomposition procedure.

Nomenclature

Below the binary variables, continuous variables, graphs, objective coefficients, parameters, sets, and subsets utilized to construct the MILP formulation of the integrated multi-period infrastructure planning and operational scheduling problem are presented.

Graphs

$G_1(V_1, E_1)$ directed graph that captures the superstructure of the energy system. The vertices, V_1, in the graph indicate the processes, storage units, and transportation mechanisms. The edges, E_1, in the graph indicate the pathways can flow between processes, storage units, and transportation mechanisms.

$G_2(V_2, E_2)$ fully connected directed graph that captures all of the possible mode transitions between components, where the vertices, V_2, are equivalent to the set of operating modes, M, and the edges, E_2, are equivalent to all the possible combinations of operating modes, $M \times M$

Sets

$\mathcal{A}$	storage units, $\{a \mid a \in \mathcal{A} \subseteq V_1\}$
$\mathcal{B}$	transportation linkages, $\{b \mid b \in \mathcal{B} \subseteq V_1\}$
$\mathcal{C}$	non-mode based processes, $\{c \mid c \in \mathcal{C} \subseteq V_1\}$
$\mathcal{D}$	mode based processes, $\{d \mid d \in \mathcal{D} \subseteq V_1\}$
$\mathcal{H}$	preexisting components, $\{h \mid h \in \mathcal{H} \subseteq V_1\}$
$\mathcal{L}$	locations, $\{l_1, l_2, \ldots, l_{\lvert\mathcal{L}\rvert}\}$
$\mathcal{P}$	planning periods, $\{p_1, p_2, \ldots, p_{\lvert\mathcal{P}\rvert}\}$
$\mathcal{R}$	commodities, $\{r_1, r_2, \ldots, r_{\lvert\mathcal{R}\rvert}\}$
$\mathcal{S}$	scheduling periods, $\{s_1, s_2, \ldots, s_{\lvert\mathcal{S}\rvert}\}$
$\mathcal{T}$	component technologies, $\{t_1, t_2, \ldots, t_{\lvert\mathcal{T}\rvert}\}$
$\mathcal{W}$	sets of representative time periods, $\{w_1, w_2, \ldots, w_{\lvert\mathcal{W}\rvert}\}$

Subsets

$E_2(d)$	edges, $E_2(d) \subseteq E_2$, that a specific component, $d \in \mathcal{D}$, can traverse
$\mathcal{R}(n)$	commodities, $\mathcal{R}(n) \subseteq \mathcal{R}$, that can be produced or consumed by a component, $n \in V_1$
$\mathcal{R}(l)$	commodities, $\mathcal{R}(l) \subseteq \mathcal{R}$, that can are available or can consumed, demanded, and or produced in a location, $l \in \mathcal{L}$
$\mathcal{R}^-(n)$	commodities, $\mathcal{R}^-(n) \subseteq \mathcal{R}$, that can be consumed by a component, $n \in V_1$
$\mathcal{R}^+(n)$	commodities, $\mathcal{R}^+(n) \subseteq \mathcal{R}$, that can be produced by a component, $n \in V_1$
$\mathcal{R}^+(l)$	commodities, $\mathcal{R}^+(l) \subseteq \mathcal{R}$, that have a strictly positive associated slack cost in a location, $l \in \mathcal{L}$.

$V_1(p)$ components, $V_1(p) \subseteq V_1$, that can be operational on or before a planning period, $p \in \mathcal{P}$

$V_1^-(n)$ components, $\{\bar{n} \in V_1 \colon (\bar{n}, n) \in E_1\}$, whose effluent commodities can flow into the component, $n \in V_1$

$V_1^+(n)$ components, $\{\bar{n} \in V_1 \colon (n, \bar{n}) \in E_1\}$, whose influent commodities can flow from the component, $n \in V_1$

$V_1^-(r)$ components, $V_1^-(r) \subseteq V_1$, that consume a resource, $r \in \mathcal{R}$

$V_1^+(r)$ components, $V_1^+(r) \subseteq V_1$, that produce a resource, $r \in \mathcal{R}$

$V_1(p, l)$ components, $V_1(p, l) \subseteq V_1$, that are positioned at a specific location, $l \in \mathcal{L}$, and can be operational on or before a planning period, $p \in \mathcal{P}$

$V_1(p, l, r)$ components, $V_1(p, l, r) \subseteq V_1$, that produced or consume, a given commodity, $r \in \mathcal{R}$, are positioned at a specific location, $l \in \mathcal{L}$, and can be operational on or before a planning period, $p \in \mathcal{P}$

$V_1^-(p, l, r)$ transportation linkages that can import a commodity, $r \in \mathcal{R}$, into a location, $l \in \mathcal{L}$, during a given planning period, $p \in \mathcal{P}$

$V_1^+(p, l, r)$ transportation linkages that can import a commodity, $r \in \mathcal{R}$, out of a location, $l \in \mathcal{L}$, during a given planning period, $p \in \mathcal{P}$

$V_2(d)$ operating modes, $V_2(d) \subseteq V_2$, that a specific component, $d \in \mathcal{D}$, can operate in

$V_2^-(d, o)$ prior operating modes, $\{\bar{o} \in V_2(d) \colon (\bar{o}, o) \in E_2(d)\}$, that a component, $d \in \mathcal{D}$, can operate in given that it is currently operating in mode, $o \in V_2(d)$

$V_2^+(d, o)$ subsequent operating modes, $\{\bar{o} \in V_2(d) \colon (o, \bar{o}) \in E_2(d)\}$, that a component, $d \in \mathcal{D}$, can operate in given that it is currently operating in mode, $o \in V_2(d)$

Parameters

$\beta_{l,r}^{p,w,s}$ nominal demand of a commodity, $r \in \mathcal{R}(l)$, a location, $l \in \mathcal{L}$, that must be met during a scheduling period, $s \in \mathcal{S}$, that belongs to a representative time period, $w \in \mathcal{W}$, and a planning period, $p \in \mathcal{P}$

$\gamma_{b,r}$ fractional amount of material, $r \in \mathcal{R}$, that is not lost via the transportation linkage, $b \in \mathcal{B}$

$\delta_{d,o,\bar{o}}^-$ maximum decrease in the operational set point of a component, $d \in \mathcal{D}$, given that it is currently in the operating mode, $\bar{o} \in V_2(d)$, and was previously in the mode, $v \in V_2^-(d, o)$

$\delta_{d,o,\bar{o}}^+$ maximum increase in the operational set point of a component, $d \in \mathcal{D}$, given that it is currently in the operating mode, $\bar{o} \in V_2(d)$, and was previously in the mode, $v \in V_2^-(d, o)$

$\epsilon_{l,r}^{p,w,s}$ maximum amount of a commodity, $r \in \mathcal{R}(l)$, that can be purchased at a location, $l \in \mathcal{L}$, during a scheduling period, $s \in \mathcal{S}$, that belongs to a representative time period, $w \in \mathcal{W}$, and a planning period, $p \in \mathcal{P}$

$\zeta_{d,o}^{-}$ minimum set point that a component, $d \in \mathcal{D}$, can operate at while in a given mode, $o \in V_2(d)$

$\zeta_{d,o}^{+}$ maximum set point that a component, $d \in \mathcal{D}$, can operate at while in a given mode, $o \in V_2(d)$

$\eta_{c,r}^{p,w,s}$ conversion factor of a non-mode based component, $c \in C \cap V_1(p,l,r)$, for a specific commodity, $r \in \mathcal{R}$, during a scheduling period, $s \in \mathcal{S}$, that belongs to a representative time period, $w \in \mathcal{W}$, and a planning period, $p \in \mathcal{P}$

$\phi_{d,o,r}^{p,w,s}$ conversion factor of a mode based component, $d \in D \cap V_1(p,l,r)$, operating in the mode, $o \in V_2(d)$, for a specific commodity, $r \in \mathcal{R}$, during a scheduling period, $s \in \mathcal{S}$, that belongs to a representative time period, $w \in \mathcal{W}$, and a planning period, $p \in \mathcal{P}$

λ_n^{-} minimum nameplate capacity of a component, $n \in V_1$

λ_n^{+} maximum nameplate capacity of a component, $n \in V_1$

$\rho_{l,r}^{p,w,s}$ maximum amount of a commodity, $r \in \mathcal{R}(l)$, that can be disposed of at a location, $l \in \mathcal{L}$, during a scheduling period, $s \in \mathcal{S}$, that belongs to a representative time period, $w \in \mathcal{W}$, and a planning period, $p \in \mathcal{P}$

$\sigma_{a,r}$ fractional amount of material, $r \in \mathcal{R}$, not lost during a scheduling period for a storage unit, $a \in \mathcal{A}$

τ_s^{-} scheduling period prior to the scheduling period, $s \in \mathcal{S}$

$\tau_{s,d,\bar{o},o}^{+}$ scheduling period subsequent to s, if the component, $d \in \mathcal{D}$, was operating in the mode, $\bar{o} \in V_2(d)$, and currently operating in the mode, $o \in V_2^{+}(d,\bar{o})$

Binary Variables

$\bar{v}_n$ 1 if the component, $n \in V_1$, is developed; otherwise, 0

$\bar{z}_{d,\bar{o},o}^{p,w,s}$ 1 if the component, $d \in \mathcal{D} \cap V_1(p)$, is operating in the mode, $o \in V_2(d)$, during a specific planning period, $p \in \mathcal{P}$, representative time period, $w \in \mathcal{W}$, scheduling period, $s \in \mathcal{S}$, and was previously operating in the mode, $\bar{o} \in V_2^{-}(d,o)$; otherwise, 0

Continuous Variables

$m_{l,r}^{p,w,s}$ amount of a commodity, $r \in \mathcal{R}(l)$, purchased at a location, $l \in \mathcal{L}$, during specific planning period, $p \in \mathcal{P}$, representative time period, $w \in \mathcal{W}$, and scheduling period, $s \in \mathcal{S}$

$n_{l,r}^{p,w,s}$ amount of a commodity, $r \in \mathcal{R}(l)$, disposed of at a location, $l \in \mathcal{L}$, during specific planning period, $p \in \mathcal{P}$, representative time period, $w \in \mathcal{W}$, and scheduling period, $s \in \mathcal{S}$

$u_n^{p,w}$ nominal operating point of a component, $n \in V_1$, during a specific planning period, $p \in \mathcal{P}$, and representative time period, $w \in \mathcal{W}$

v_n nameplate capacity of component, $n \in V_1$

$x_n^{p,w,s}$ operational set point of a component, $n \in V_1(p) \setminus \mathcal{D}$, during a specific planning period, $p \in \mathcal{P}$, representative time period, $w \in \mathcal{W}$, and scheduling period, $s \in \mathcal{S}$

$y_{d,o}^{p,w,s}$ operational set point of a component, $d \in \mathcal{D} \cap V_1(p)$, if it operating in the mode, $o \in V_2(d)$, during a specific planning period, $p \in \mathcal{P}$, representative time period, $w \in \mathcal{W}$, and scheduling period, $s \in \mathcal{S}$

Objective Coefficients

$\Delta_{l,r}^{p,w,s}$ variable operational cost to purchase a commodity, $r \in \mathcal{R}(l)$, at a location, $l \in \mathcal{L}$, during specific planning period, $p \in \mathcal{P}$, representative time period, $w \in \mathcal{W}$, and scheduling period, $s \in \mathcal{S}$

$P_{l,r}^{p,w,s}$ variable operational cost to dispose of a commodity, $r \in \mathcal{R}(l)$, at a location, $l \in \mathcal{L}$, during specific planning period, $p \in \mathcal{P}$, representative time period, $w \in \mathcal{W}$, and scheduling period, $s \in \mathcal{S}$

V_n variable capital cost of a component, $n \in V_1 \setminus \{\mathcal{D} \cup \mathcal{H}\}$

$X_n^{p,w,s}$ variable operational cost to operate a component, $n \in V_1(p) \setminus \mathcal{D}$, during a specific planning period, $p \in \mathcal{P}$, representative time period, $w \in \mathcal{W}$, and scheduling period, $s \in \mathcal{S}$

$Y_{d,o}^{p,w,s}$ variable operational cost to operate a component, $d \in \mathcal{D} \cap V_1(p)$, if it is operating in the mode, $o \in V_2(d)$, during a specific planning period, $p \in \mathcal{P}$, representative time period, $w \in \mathcal{W}$, and scheduling period, $s \in \mathcal{S}$

$\bar{V}_n$ fixed capital cost of a component, $n \in V_1 \setminus \mathcal{H}$

$\bar{Y}_{d,\bar{o},o}^{p,w,s}$ fixed operational cost to operate a component, $d \in \mathcal{D} \cap V_1(p)$, if it is operating in the mode, $o \in V_2(d)$, during a specific planning period, $p \in \mathcal{P}$, representative time period, $w \in \mathcal{W}$, and scheduling period, $s \in \mathcal{S}$, and was previously operating in the mode, $\bar{o} \in V_2^-(d, o)$

Present Addresses

R. Cory Allen is currently located at the ExxonMobil Upstream Research Company in Spring, TX, USA.

Acknowledgements

This research was funded by: (i) Royal Dutch Shell; (ii) the National Science Foundation under grant no. 1739977 (INFEWS); and (iii) the Texas A&M Energy Institute.

References

[1] R. C. Allen, Y. Nie, S. Avraamidou, and E. N. Pistikopoulos. Infrastructure planning and operational scheduling for power generating systems: An energy-water nexus approach. In *Computer Aided Chemical Engineering*, volume 47, pages 233–238. Elsevier, 2019.

[2] R. C. Allen, S. G. Baratsas, R. Kakodkar, S. Avraamidou, J. B. Powell, C. F. Heuberger, C. D. Demirhan, and E. N. Pistikopoulos. An optimization framework for solving integrated planning and scheduling problems for dense energy carriers. *IFAC-PapersOnLine*, 54(3):621–626, 2021.

[3] R. C. Allen, S. G. Baratsas, R. Kakodkar, S. Avraamidou, C. D. Demirhan, C. F. Heuberger-Austin, M. Klokkenburg, and E. N. Pistikopoulos. A multi-period integrated planning and scheduling approach for developing energy systems. *Optimal Control Applications and Methods*, 2022.

[4] R. C. Allen, C. D. Demirhan, C. F. Heuberger-Austin, and E. N. Pistikopoulos. A multi-period planning and scheduling strategy for developing hydrogen-based supply chains. In *Computer Aided Chemical Engineering*, volume 51, pages 919–924. Elsevier, 2022.

[5] A. Allman, M. J. Palys, and P. Daoutidis. Scheduling-informed optimal design of systems with time-varying operation: A wind-powered ammonia case study. *AIChE Journal*, 65(7):e16434, 2019.

[6] P. Baptiste, C. Le Pape, and W. Nuijten. *Constraint-based scheduling: applying constraint programming to scheduling problems*, volume 39. Springer Science & Business Media, 2001.

[7] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Computational Management Science*, 2(1):3–19, 1962.

[8] C. D. Demirhan, W. W. Tso, J. B. Powell, C. F. Heuberger, and E. N. Pistikopoulos. A multiscale energy systems engineering approach for renewable power generation and storage optimization. *Industrial & Engineering Chemistry Research*, 59(16):7706–7721, 2020.

[9] C. D. Demirhan, W. W. Tso, J. B. Powell, and E. N. Pistikopoulos. A multi-scale energy systems engineering approach towards integrated multi-product network optimization. *Applied Energy*, 281:116020, 2021.

[10] C. Draxl, A. Clifton, B.-M. Hodge, and J. McCaa. The wind integration national dataset (wind) toolkit. *Applied Energy*, 151:355–366, 2015.

[11] I. Dunning, J. Huchette, and M. Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017. doi: 10.1137/15M1020575.

[12] V. Goel, K. C. Furman, J.-H. Song, and A. S. El-Bakry. Large neighborhood search for lng inventory routing. *Journal of Heuristics*, 18(6):821–848, 2012.

[13] M. Hummon, E. Ibanez, G. Brinkman, and D. Lew. Sub-hour solar data for power system modeling from static spatial variability analysis. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2012.

[14] R. Kakodkar, G. He, C. Demirhan, M. Arbabzadeh, S. Baratsas, S. Avraamidou, D. Mallapragada, I. Miller, R. Allen, E. Gençer, et al. A review of analytical and optimization methodologies for transitions in multi-scale energy systems. *Renewable and Sustainable Energy Reviews*, 160:112277, 2022.

[15] C. L. Lara, D. S. Mallapragada, D. J. Papageorgiou, A. Venkatesh, and I. E. Grossmann. Deterministic electric power infrastructure planning: Mixed-integer programming model and nested decomposition algorithm. *European Journal of Operational Research*, 271(3):1037–1054, 2018.

[16] C. Li, A. J. Conejo, P. Liu, B. P. Omell, J. D. Siirola, and I. E. Grossmann. Mixed-integer linear programming models and algorithms for generation and transmission expansion planning of power systems. *European Journal of Operational Research*, 297(3):1071–1082, 2022.

[17] C. Li, A. J. Conejo, J. D. Siirola, and I. E. Grossmann. On representative day selection for capacity expansion planning of power systems under extreme operating conditions. *International Journal of Electrical Power & Energy Systems*, 137:107697, 2022.

[18] Z. Li and M. G. Ierapetritou. Integrated production planning and scheduling using a decomposition framework. *Chemical Engineering Science*, 64(16):3585–3597, 2009.

[19] C. T. Maravelias and C. Sung. Integration of production planning and scheduling: Overview, challenges and opportunities. *Computers & Chemical Engineering*, 33(12):1919–1930, 2009.

[20] M. J. Palys and P. Daoutidis. Using hydrogen and ammonia for renewable energy storage: A geographically comprehensive techno-economic study. *Computers & Chemical Engineering*, 136:106785, 2020.

[21] M. J. Palys and P. Daoutidis. Power-to-x: A review and perspective. *Computers & Chemical Engineering*, page 107948, 2022.

[22] M. J. Palys, I. Mitrai, and P. Daoutidis. Renewable hydrogen and ammonia for combined heat and power systems in remote locations: Optimal design and scheduling. *Optimal Control Applications and Methods*, 2021.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[24] N. K. Shah and M. G. Ierapetritou. Integrated production planning and scheduling optimization of multisite, multiproduct process industry. *Computers & Chemical Engineering*, 37: 214–226, 2012.

[25] Y. Shao, K. C. Furman, V. Goel, and S. Hoda. A hybrid heuristic strategy for liquefied natural gas inventory routing. *Transportation Research Part C: Emerging Technologies*, 53:151–171, 2015.

[26] Gurobi Optimization, LLC. Gurobi optimizer reference manual - version 9.5, 2022.

[27] Shell. Shell scenarios sketch: A US net-zero CO_2 energy system by 2050, 2020.

[28] J. J. Torres, C. Li, R. M. Apap, and I. E. Grossmann. A review on the performance of linear and mixed integer two-stage stochastic programming algorithms and software. *Optimization Online*, 2019.

[29] S. D. Tsolas and M. FaruqueHasan. Resilience and survivability-aware optimal design and operation of interconnected supply chains. In *Computer Aided Chemical Engineering*, volume 50, pages 549–554. Elsevier, 2021.

[30] S. D. Tsolas and M. F. Hasan. Survivability-aware design and optimization of distributed supply chain networks in the post covid-19 era. *Journal of Advanced Manufacturing and Processing*, 3(3):e10098, 2021.

[31] Q. Zhang, M. Martin, and I. E. Grossmann. Integrated design and operation of renewables-based fuels and power production networks. *Computers & Chemical Engineering*, 122:80–92, 2019.

Supplementary Material

Problem Formulation

In this subsection, the MILP formulation, P, utilized to model the integrated multi-period infrastructure planning and operational scheduling problem is presented. This MILP formulation is a reduction of an earlier developed MIQP formulation utilizing for solving integrated multi-period infrastructure planning and operational scheduling problems [3]. The reduction arises from the fact that in this formulation the nameplate capacities of the mode-based process are selected a priori – it should be noted, that while the nameplate capacities are known, whether or not the processes are constructed is not known. This is in turn, removes the need for the bi-linear terms that occur in the aforementioned MIQP problem.

$$P = \min\ J_3 + J_4$$

$$\text{s.t.}\ \sum\nolimits_{a\in\mathcal{A}\cap V_1(p,l,r)} \sigma_{a,r}\cdot x_a^{p,w,\tau_s^-} + \sum\nolimits_{b\in V_1^-(p,l,r)} \gamma_{b,r}^s \cdot x_b^{p,w,\tau_{s,b}^-} + \ldots \tag{S1}$$

$$\sum\nolimits_{c\in\mathcal{C}\cap V_1(p,l,r)} \eta_{c,r}^{p,w,s}\cdot x_c^{p,w,s} + \sum\nolimits_{d\in\mathcal{D}\cap V_1(p,l,r)} \sum\nolimits_{o\in V_2(d)} \phi_{d,o,r}^{p,w,s}\cdot y_{d,o}^{p,w,s} + \ldots$$

$$m_{l,r}^{p,w,s} = \beta_{l,r}^{p,w,s} + n_{l,r}^{p,w,s} + \sum\nolimits_{b\in V_1^+(p,l,r)} x_b^{p,w,s} + \sum\nolimits_{a\in\mathcal{A}\cap V_1(p,l,r)} x_a^{p,w,s}\ \forall\ p\in\mathcal{P},\ \ldots$$

$$w\in\mathcal{W},\ s\in\mathcal{S},\ l\in\mathcal{L},\ r\in\mathcal{R}(l)$$

$$\sum\nolimits_{\bar{o}\in V_2^-(d,o)} \bar{z}_{d,\bar{o},o}^{p,w,s} = \sum\nolimits_{\bar{o}\in V_2^+(d,o)} \bar{z}_{d,o,\bar{o}}^{p,w,\tau_{s,d,o,\bar{o}}^+}\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ s\in\mathcal{S},\ \ldots \tag{S2}$$

$$d\in\mathcal{D}\cap V_1(p)\, o\in V_2(d)$$

$$\sum\nolimits_{(\bar{o},o)\in E_2(d)} \bar{z}_{d,\bar{o},o}^{p,w,s} \le 1\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ s\in\mathcal{S},\ d\in\mathcal{D}\cap V_1(p) \tag{S3}$$

$$y_{d,o}^{p,w,s} \ge \zeta_{d,o}^- \cdot \sum\nolimits_{\bar{o}\in V_2^-(d,o)} \bar{z}_{d,\bar{o},o}^{p,w,s}\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ s\in\mathcal{S},\ d\in\mathcal{D}\cap V_1(p),\ o\in V_2(d) \tag{S4}$$

$$y_{d,o}^{p,w,s} \le \zeta_{d,o}^+ \cdot \sum\nolimits_{\bar{o}\in V_2^-(d,o)} \bar{z}_{d,\bar{o},o}^{p,w,s}\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ s\in\mathcal{S},\ d\in\mathcal{D}\cap V_1(p),\ o\in V_2(d) \tag{S5}$$

$$y_{d,o}^{p,w,s} - y_{d,\bar{o}}^{p,w,\tau_{s,d,\bar{o},o}^-} \le (\delta_{d,\bar{o},o}^+ - \zeta_{d,o}^+)\cdot \bar{z}_{d,\bar{o},o}^{p,w,s} + \zeta_{d,o}^+\cdot \bar{v}_d\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ s\in\mathcal{S},\ \ldots \tag{S6}$$

$$d\in\mathcal{D}\cap V_1(p),\ (\bar{o},o)\in E_2(d)$$

$$y_{d,o}^{p,w,s} - y_{d,\bar{o}}^{p,w,\tau_{s,d,\bar{o},o}^-} \ge (\delta_{d,\bar{o},o}^- + \zeta_{d,\bar{o}}^+)\cdot \bar{z}_{d,\bar{o},o}^{p,w,s} - \zeta_{d,\bar{o}}^+\cdot \bar{v}_d\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ s\in\mathcal{S},\ \ldots \tag{S7}$$

$$d\in\mathcal{D}\cap V_1(p),\ (\bar{o},o)\in E_2(d)$$

$$x_n^{p,w,s} \le v_n\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ s\in\mathcal{S},\ n\in V_1(p)\setminus\mathcal{D} \tag{S8}$$

$$v_n \ge \lambda_n^-\cdot\bar{v}_n\ \forall\ n\in V_1 \tag{S9}$$

$$v_n \le \lambda_n^+\cdot\bar{v}_n\ \forall\ n\in V_1 \tag{S10}$$

$$m_{l,r}^{p,w,s}\in[0,\epsilon_{l,r}^{p,w,s}]\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ s\in\mathcal{S},\ l\in\mathcal{L},\ r\in\mathcal{R}(l) \tag{S11}$$

$$n_{l,r}^{p,w,s}\in[0,\rho_{l,r}^{p,w,s}]\ \forall\ p\in\mathcal{P},\ w\in\mathcal{W},\ s\in\mathcal{S},\ l\in\mathcal{L},\ r\in\mathcal{R}(l) \tag{S12}$$

$$v_n\in[0,\lambda_n^+]\ \forall\ n\in V_1 \tag{S13}$$

$$x_n^{p,w,s} \in [0, \lambda_n^+] \; \forall \; p \in \mathcal{P},\; w \in \mathcal{W},\; s \in \mathcal{S},\; n \in V_1(p) \setminus \mathcal{D} \tag{S14}$$

$$y_{d,o}^{p,w,s} \in [0, \lambda_d^+] \; \forall \; p \in \mathcal{P},\; w \in \mathcal{W},\; s \in \mathcal{S},\; d \in \mathcal{D} \cap V_1(p),\; o \in V_2(d) \tag{S15}$$

$$\bar{v}_n \in \{0, 1\} \; \forall \; n \in V_1 \tag{S16}$$

$$\bar{z}_{d,\bar{o},o}^{p,w,s} \in \{0, 1\} \; \forall \; p \in \mathcal{P},\; w \in \mathcal{W},\; s \in \mathcal{S},\; d \in \mathcal{D} \cap V_1(p),\; (\bar{o}, o) \in E_2(d) \tag{S17}$$

The objective functions, J_3 and J_4, in P are given by Eqs. (S18) and (S19). Equation (S18) sums the variable cost to: (i) dispose of excess commodities, raw materials, intermediate products, and or final products, at the the commodity sinks in the scheduling problems; (ii) purchase additional commodities, raw materials, intermediate products, and or final products, at the the commodity sources in the scheduling problems; (iii) construct a component in the energy system; (iv) operate the non-mode based components in the scheduling problems; and (v) operate the mode based components in the scheduling problems. Equation (S19) sums the fixed cost to: (i) construct a mode based component in the energy system; and (ii) operate the mode based components in the scheduling problems.

$$\begin{aligned} J_3 = & \sum\nolimits_{p \in \mathcal{P}} \sum\nolimits_{w \in \mathcal{W}} \sum\nolimits_{s \in \mathcal{S}} \sum\nolimits_{l \in \mathcal{L}} \sum\nolimits_{r \in \mathcal{R}(l)} \Delta_{l,r}^{p,w,s} \cdot m_{l,r}^{p,w,s} + \ldots \\ & \sum\nolimits_{p \in \mathcal{P}} \sum\nolimits_{w \in \mathcal{W}} \sum\nolimits_{s \in \mathcal{S}} \sum\nolimits_{l \in \mathcal{L}} \sum\nolimits_{r \in \mathcal{R}(l)} P_{l,r}^{p,w,s} \cdot n_{l,r}^{p,w,s} + \ldots \\ & \sum\nolimits_{n \in V_1 \setminus \{\mathcal{D} \cup \mathcal{H}\}} V_f \cdot v_f + \ldots \\ & \sum\nolimits_{p \in \mathcal{P}} \sum\nolimits_{w \in \mathcal{W}} \sum\nolimits_{s \in \mathcal{S}} \sum\nolimits_{n \in V_1(p) \setminus \mathcal{D}} X_n^{p,w,s} \cdot x_n^{p,w,s} + \ldots \\ & \sum\nolimits_{p \in \mathcal{P}} \sum\nolimits_{w \in \mathcal{W}} \sum\nolimits_{s \in \mathcal{S}} \sum\nolimits_{d \in \mathcal{D} \cap V_1(p)} \sum\nolimits_{o \in V_2(d)} Y_{d,o}^{p,w,s} \cdot y_{d,o}^{p,w,s} \end{aligned} \tag{S18}$$

$$\begin{aligned} J_4 = & \sum\nolimits_{n \in V_1 \setminus \mathcal{H}} \overline{V}_{2n} \cdot \bar{v}_n + \ldots \\ & \sum\nolimits_{p \in \mathcal{P}} \sum\nolimits_{w \in \mathcal{W}} \sum\nolimits_{s \in \mathcal{S}} \sum\nolimits_{d \in \mathcal{D} \cap V_1(p)} \sum\nolimits_{(\bar{o}, o) \in E_2(d)} \bar{Z}_{d,\bar{o},o}^{p,w,s} \cdot \bar{z}_{d,\bar{o},o}^{p,w,s} \end{aligned} \tag{S19}$$

The first set of constraints, Eq. (S1), in the ensure that the conservation of mass holds in the scheduling problems between sequential scheduling periods for each planning period, representative scheduling scenario, scheduling period, location in the energy system, and commodity. The first set of summations in Eq. (S1), $\sum_{a \in \mathcal{A} \cap V_1(p,l,r)} \sigma_{a,r} \cdot x_a^{p,w,\tau_s^-}$, account for the commodities that were in storage in the previous scheduling period. The second set of summations in Eq. (S1), $\sum_{b \in V_1^-(p,l,r)} \gamma_{b,r}^s \cdot x_b^{p,w,\tau_{s,b}^-}$, account for the commodities that are imported into the location in the energy system currently under consideration. The third set of summations in Eq. (S1), $\sum_{c \in \mathcal{C} \cap V_1(p,l,r)} \eta_{c,r}^{p,w,s} \cdot x_c^{p,w,s}$, account for the commodities produced and or consumed by the non-mode based processes. The fourth set of summations in Eq. (S1), $\sum_{d \in \mathcal{D} \cap V_1(p,l,r)} \sum_{o \in V_2(d)} \phi_{d,o,r}^{p,w,s} \cdot y_{d,o}^{p,w,s}$, account for the commodities produced and or consumed by the mode based processes. The fifth set of summations in Eq. (S1), $\sum_{b \in V_1^+(p,l,r)} x_b^{p,w,s}$, account for the commodities that are exported

out of the current location under consideration. The sixth and final set of set of summations in Eq. (S1), $\sum_{a\in\mathcal{A}\cap V_1(p,l,r)} x_a^{p,w,s}$, account for the commodities that are in storage in the current scheduling period.

The dynamics of the mode based processes have been captured by a cyclic "time-state" network flow formulation, please see Example 1 for a brief illustration of the underlying concepts or the earlier works [3, 12, 25]. The second set of constraints given by Eq. (S2) ensure that the proper sequence of mode transitions between for the processes is followed throughout the scheduling problems. The third set of constraints given by Eq. (S3), ensure that a mode based process can only be in at most one operating mode during a scheduling period. The fourth and fifth sets of constraints given by Eq. (S4) and Eq. (S5) respectively, ensure that a mode based process is must operate between its operational bounds. The sixth and seventh sets of constraints given by Eq. (S6) and Eq. (S7) respectively, bound the changes in set points between sequential scheduling periods and operating modes.

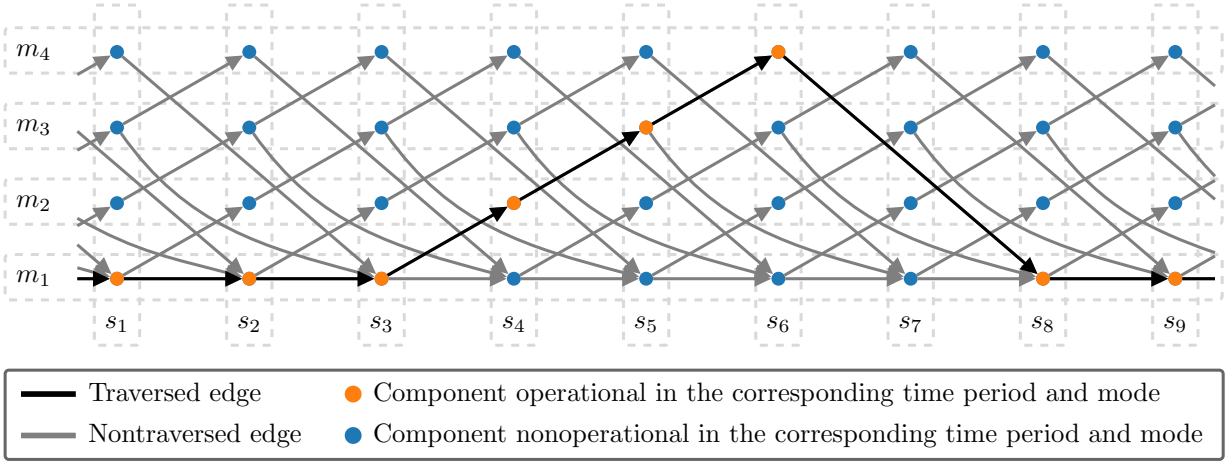

Figure 6: "Time-state" network representation utilized to capture the complex mode based process dynamics.

Illustrative Example 1 *To elucidate the concept of mode based process dynamics, assume that a component in the energy system has a minimum down time of 2 scheduling periods, a minimum up time of 2 scheduling periods, and a maximum up time of 3 scheduling periods. Figure 6 is a "time-state" network representation that captures the mode based transitions of the aforementioned process when the scheduling horizon,* $\{s_1, s_2, \ldots, s_9\}$. *In this example there are 4 operating modes,* $\{m_1, m_2, m_3, m_4\}$. *The first operating mode,* m_1, *is utilized to indicate that the component is "offline", the second operating mode,* m_2, *is utilized to indicate that the component has been "online" for 1 scheduling period, the third operating mode,* m_3, *is utilized to indicate that the component has been "online" for 2 scheduling periods, and the fourth and final operating mode,* m_4, *is utilized to indicate that the component has been "online" for 3 scheduling periods. The edges in the network representation ensure that the component is "offline" for a minimum of 2 scheduling periods, "online" for a minimum of 2 scheduling periods, and "online" for a maximum of 3 scheduling*

periods.

In this contrived example, assume that the component is "offline", for the first three time periods, s_1, s_2, and s_3, "online" for the next three subsequent time periods, s_4, s_5, and s_6, and then "offline", for the final three time periods, s_7, s_8, and s_9. In Fig. 6, the black edges and orange nodes illustrates this behavior.

The eighth set of constraints given by Eq. (S8), acts as an upper bound on the operating set point of the non-mode based components in the energy system. The ninth and tenth set of constraints given by Eq. (S9) and Eq. (S10) bound the nameplate capacity of the components in the energy system. It should be highlighted, if the component, $n \in V_1$, under consideration is a mode-based process then $\lambda_n^+ = \lambda_n^-$, which ensures that the nameplate capacity of the component is either 0 or λ_n^+.

Equations (S11) and (S12) describe the continuous variables that are utilized to track the amount of commodities disposes at a commodity sinks and the amount of commodities purchased from the commodity sources. Equation (S13) describes the continuous variables that are utilized to track the nameplate plate capacities of the components in the energy system. Equations (S14) and (S15) describe the continuous variables that are utilized to track the operating set points of the non-mode based components and mode-based components respectively. Equation (S16) describes the binary variables that are utilized to track if a component utilized in the energy system or not. Equation (S17) describes the binary variables that are utilized to operating modes of the mode based processes.