Novel Multiple Time-grid Continuous-time Mathematical Formulation for Short-term Scheduling of Multipurpose Batch Plants

Dan Li, Nikolaos Rakovitis, Taicheng Zheng, Yueting Pan, Jie Li,* and Giorgos Kopanos

ABSTRACT: In this work, we have developed two novel unit-specific event-based mixed-integer linear programming models for scheduling multipurpose batch plants. The concept of indirect and direct material transfer is introduced to rigorously sequence and align tasks in different units. A batch after production is allowed to be partially transferred to storage and downstream processing units or held in processing units over multiple event points. The computational results demonstrate that the proposed models require a smaller number of event points in many cases to achieve optimality than existing unit-specific event-based models. It is interesting to find that no task is required to span over multiple event points to reach optimality for all addressed examples. The best variant developed is superior to existing unit-specific event-based models with the same or better objective values by a maximum improvement of 67%. The computational effort is significantly reduced by at least 1 order of magnitude in some cases.

1. INTRODUCTION

Multipurpose batch plants widely exist in the chemical industry for production of a large number of low-volume, high-value products. To achieve higher utilization of resources, lower inventory costs, and better responsiveness to a fluctuating manufacturing environment, optimal scheduling of multipurpose facilities is desirable and has attracted much interest of both academia and industry in the past decades. Many mathematical models have been developed based on either a state task network (STN) or a resource task network (RTN). These models can be classified into discrete- and continuous-time models. The continuous-time models can be further classified into global event-based, slot-based including process-slot based and unit-slot based, and sequence-based models. These models can also be classified into single- and multiple-time grid models. For more details about these models, the readers are referred to the excellent review papers.

All existing time-grid mathematical models divide the scheduling horizon using time points/slots/event points on which a task, operation, or activity can start, finish, or both. Therefore, the number of time points/slots/event points directly affects the efficiency of these mathematical models. More specifically, an increment in the number of time points/slots/event points can lead to an exponential increase in the model size (i.e., the number of binary variables, continuous variables, and constraints), which can potentially increase the computational time by at least 1 order of magnitude to generate the optimal solution. To reduce the required number of time points/slots/event points, Seid and Majozi observed that if there are enough materials in their dedicated storage for consumption, some sequence constraints related to different tasks in different units could be relaxed. They also implied that if there is sufficient storage space to hold materials, all related production and consumption tasks are not necessary to be rigorously aligned. Although their formulation did require a smaller number of event points in some cases, it could generate a schedule with real-time violation. Vooradi and Shaik noticed that if a consumption task consumes a state from a specifically related production task, this consumption task should start only after the specifically related production task instead of all related production tasks like Seid and Majozi. They also required related production and consumption tasks not to be rigorously aligned for sufficient storage space. They improved their previous formulation of Shaik and Floudas, which indeed further reduced the number of event points required without real-time violation in comparison to the model of Seid and Majozi. However, they introduced an increased number of binary variables, leading to computational effort.

Received: April 18, 2022
Revised: September 16, 2022
Accepted: September 27, 2022
Published: October 19, 2022
inefficiency. In addition, their model can lead to a greater number of event points in many cases.

Besides the abovementioned two attempts, Vooradi and Shaik\textsuperscript{25} and Mostafaei and Harjunkoski\textsuperscript{26} attempted to eliminate some big-M constraints in their developed models, which could lead to much more slots/event points required and sub-optimality. More importantly, the multi-grid model of Mostafaei and Harjunkoski\textsuperscript{26} is applicable only when the storage capacity is unlimited. With limited or finite storage capacities, the schedules obtained from their models are often infeasible due to violation of storage capacities. Shaik and Vooradi\textsuperscript{27} and Rakovitis et al.\textsuperscript{19} imposed related production and consumption tasks to start at the same events. To improve the schedule accuracy for varying processing times from the discrete-time models, the combination of discrete- and continuous-time formulations was also investigated.\textsuperscript{28,29} In addition, machine learning techniques were also applied for online production scheduling.\textsuperscript{30,31}

In this work, we mainly focus on the existing multiple-time-grid continuous-time models, especially the unit-specific event-based models for short-term scheduling as their capabilities are well established in the literature.\textsuperscript{13,14,19} They are also the basis for the development of decomposition algorithms to solve industrial-scale scheduling problems. It is found that almost all existing unit-specific event-based models\textsuperscript{16,17} for scheduling of multipurpose batch plants still require a great number of event points in many cases, leading to computational inefficiency. More importantly, almost all existing models did not allow a batch of materials to be temporarily stored in processing units or be partially transferred to downstream units after production, leading to sub-optimality and inefficient utilization of processing units. Therefore, in this work, we have developed two novel multiple-time-grid continuous-time formulations using the unit-specific event-based modeling approach to address the abovementioned limitations. The concept of indirect and direct material transfer is explicitly introduced to sequence different tasks in different units, which allow material transfer to be monitored between units rather than specific tasks to reduce the number of discrete decision variables and improve the computational efficiency. A batch of materials after production is allowed to be partially transferred to storage and downstream processing units or temporarily held in suitable processing units over multiple event points. A new continuous variable denoting the time when a state produced in a processing unit is available at an event point is introduced to assist the sequencing of different tasks related to the same states in different units, which overcomes the limitations in the existing unit-specific event-based models.\textsuperscript{17}

To further reduce the computational expense, some new tightening constraints are proposed. A number of benchmark examples are solved to evaluate the performance of the proposed formulations. The computational results demonstrate that the best variant developed is superior to the existing unit-specific event-based models.

2. PROBLEM STATEMENT

Figure 1 illustrates an STN representation of a typical multipurpose batch plant, where batch splitting, mixing, and recycling are permitted. There are $J$ tasks such as heating, reaction, and separation processed in $J$ units. The unit $j$ can process $I_j$ tasks. However, it can process at most one task at a time. All materials including raw materials $S^0$, intermediate products $S^m$, and final products $S^f$ are denoted as states. There are $S$ states, related $I$ tasks, initial amount, and storage capacities; (c) production recipes (i.e., the coefficients of processing time for each task, and the consuming or producing fractions of tasks); (d) economic data (e.g., product price, raw material cost, and unit operating cost), scheduling horizon and product demands. It is to determine (a) optimal production schedule including allocation, sequence, timings of tasks in a unit; (b) amount of materials being processed in each unit at each time; (c) inventory profiles of all material states. Assumptions include (i) all parameters are deterministic without batch/unit failures or operational interruptions; (ii) the processing time of a task depends on the batch size ($b_j$), which is represented by $\alpha_i + \beta_i b_j$; (iii) limited feed materials and other resources like utility and manpower; (iv) unlimited storage for raw materials and final products; (v) negligible transfer time between facilities (e.g., units and storage tanks); (vi) setup or changeover time is lumped into processing time; (vii) each state has its dedicated storage.

Two optimization objectives are considered, including maximization of profit over a given scheduling horizon and minimization of makespan to fulfill consumer demands. The detailed problem description can be referred to Li and Floudas.\textsuperscript{14}

3. MOTIVATING EXAMPLES

We create three examples to illustrate the limitations of existing time-grid continuous-time models for scheduling multipurpose batch plants. The STN representations and relevant data are provided in Figures S2, S3, and Tables S1–S5 in the Supporting Information.

3.1. Motivating Example 1. In this example, state S2 is produced by task T1 in unit J1 and consumed by task T2 in unit J2. It has a dedicated storage with a maximum capacity of 10 mu (mass units) and no initial inventory. S3 is the final product priced at $5$ mu$^{-1}$. The objective is to maximize profit over a given scheduling horizon of 8 h. We use the models of Susarla, Li, and Karimi,\textsuperscript{10} Li and Floudas,\textsuperscript{14} and Vooradi and Shaik\textsuperscript{17} (denoted as SLK2, L&F, and V&S, respectively) to solve this example. These models are excellent representations.
of existing multiple time-grid continuous-time models. Note that the multiple time-grid model of Mostafaei and Harjunkoski\textsuperscript{26} is not used as it is not suitable for limited storage capacities. The schedule with a maximum profit of $300 obtained from these models is illustrated in Figure 2a, where colors are used to represent different tasks and numbers in parentheses indicate the batch size of tasks (the same for all figures in this paper). However, we can generate another

![Figures 2, 3, and 4 showing schedules for motivating examples.](https://doi.org/10.1021/acs.iecr.2c01363)
schedule (see Figure 2b) with a maximum profit of $500. It indicates that a better objective value can be obtained than that from all the abovementioned models,$^{10,14,17}$ (hence all existing multiple time-grid continuous-time models) due to the fact that in Figure 2b S0 mu of S2 produced in unit J1 is held in J1 at event point N2 from 5 to 6.5 h, while the rest 50 mu of S2 is transferred into unit J2 for further processing. In other words, a batch of S2 is partially transferred to storage or downstream processing units instead of being fully transferred.  

3.2. Motivating Example 2. This example is almost the same as the motivating example 1 except with the fixed processing times for tasks, as listed in Table S2. The objective is to minimize makespan with the demand of S3 being 100 mu. Besides the abovementioned representative multiple time-grid continuous-time models,$^{10,14,17}$ a representative discrete-time model proposed by Velez, Merchan, and Maravelias (referred as VMM) is used to solve this example. All these representative models generate a best schedule with a makespan of 11.5 h (see Figure 3a). A better makespan of 8 h can be generated if the batch of S2 produced by task T1 is allowed to be partially transferred to unit J2 and partially held in unit J1 from 5 to 6.5 h (Figure 3b).

3.3. Motivating Example 3. In this example, five intermediate states (S1–S5) are subject to FIS with a maximum capacity of 1000 mu and no initial inventory. Demands for products P1 and P2 are 100 and 200 mu, respectively. The objective is to minimize makespan. The optimal makespan of 19 h is generated from V&S using 18 event points with $\Delta n = 6$, L&F using 18 event points with $\Delta n = 0$, SLK2 with 11 slots (12 slot points) and VMM. Note that $\Delta n$ is a parameter denoting the maximum number of event points over which a task is allowed to span. The optimal schedule from V&S is illustrated in Figure 4a. However, the same makespan can be generated manually only using 10 event points without allowing a task to span across multiple event points (see Figure 4b). The number of event points required is reduced by 44%, compared to V&S and L&F.

All these examples motivate us to develop a new mathematical model to address the abovementioned limitations of existing multiple time-grid continuous-time models especially the unit-specific event-based models, which may lead to a significant increase in solution quality.

4. MATHEMATICAL FORMULATIONS

It is of great importance to represent the time horizon before developing a mathematical formulation. Although there are several existing time representations,$^4,23,24$ the well-established unit-specific event-based time representation$^{17–19}$ is adopted in this work (see Figure S1 in the Supporting Information) as it is more flexible and efficient (e.g., smaller model size and less computational efforts), compared to other continuous-time representations. It is often more efficient and accurate than the discrete-time variant in handling variable processing times.$^{26}$ For more details about this time representation, the readers can refer to Ierapetritou and Floudas.$^{12}$

4.1. Model M1. We introduce four-index binary variables $w_{ijn} \nu$ to denote the allocation of tasks to units

$$w_{ijn} \nu = \begin{cases} 1 & \text{if task } i \text{ is processed in unit } j \text{ from event point } n \text{ to } n' \\ 0 & \text{otherwise} \end{cases}$$

where $n$ and $n'$ are event points with $n \leq n' \leq n + \Delta n$. The parameter $\Delta n$ denotes the maximum number of event points that a task is allowed to span over. As discussed before, a state is allowed to be held in a processing unit over consecutive event points after production, if it is neither unstable nor subject to $ZW$ policy. We introduce new binary variables $y_{ijn}$ to denote holding operations as follows

$$y_{ijn} = \begin{cases} 1 & \text{if a batch previously produced by task } i \text{ is held in unit } j \text{ at } n \\ 0 & \text{otherwise} \end{cases}$$

4.1.1. Allocation Constraints. At most, one task is allowed to be processed or temporarily held in a processing unit $j$ at a time.

$$\sum_{i \in I_j} \sum_{i \leq n' \leq n} \sum_{n \leq t \leq n + \Delta n} w_{ijn} \nu + \sum_{i \in I_j, n' = n + \Delta n} y_{ijn} \leq 1 \ \forall j, n$$

(1)

where set $I^p$ denotes tasks producing states that can be temporally held in processing units.

4.1.2. Capacity Constraints. The batch size ($b_{ijn}$) should be within the minimum ($B_{ijn}^{min}$) and maximum ($B_{ijn}^{max}$) capacities.

$$B_{ijn}^{min} \cdot w_{ijn} \nu \leq b_{ijn} \leq B_{ijn}^{max} \cdot w_{ijn} \nu \ \forall j, i \in I_j, n \leq n' \leq n + \Delta n$$

(2)

4.1.3. Material Balance Constraints. We define positive variables $b_{ijn}$ to indicate the amount of a batch from task $i$ that is held in unit $j$ at event point $n$. If all states produced by task $i$ are subject to UIS, it is unnecessary to hold some amount of the batch after production. In other words, $y_{ijn} = b_{ijn} \nu = 0$ for UIS. Positive variables $ST_{ijn}^{M1}$ are defined to denote the total amount of a state $s$ stored at event point $n$ including the amount stored in storage and processing units. The amount of state $s$ stored at the beginning of event point $n$ should be equal to the amount stored at the beginning of event point $(n - 1)$, plus the amount produced at the end of event point $(n - 1)$, minus the amount consumed at the beginning of event point $n$ (see eqs 3 and 4).

$$ST_{ijn}^{M1} = ST_{ijn}^{M1, (n-1)} + \sum_{i \in I_j} \sum_{i \leq n' \leq n + \Delta n} \sum_{i \leq n' \leq n + \Delta n} \rho_{ijn} \cdot b_{ijn} \nu$$

$$\forall j, i \in I_j, n \leq n' \leq n + \Delta n$$

(3)

$$ST_{ijn}^{M1} = ST_{ijn}^{M1} + \sum_{i \in I_j, n' = n + \Delta n} \sum_{i \leq n' \leq n + \Delta n} \rho_{ijn} \cdot b_{ijn} \nu$$

$$\forall j, i \in I_j, n \leq n' \leq n$$

(4)

where $bs_{0j}$ is the initial amount of a batch held in unit $j$ produced by task $i$, $ST_{0j}$ is the initial inventory level of state $s$. When $\rho_{ijn} > 0$, task $i$ produces state $s$ (called production tasks of $s$ included in set $I^p_j$), while $\rho_{ijn} < 0$, task $i$ consumes state $s$ (called consumption tasks of $s$ included in set $I^p_j$).
The amount of state $s$ held in processing units should not exceed its total amount stored.

$$\sum_j \sum_{i \in (1, N)} \beta_{i,j} s_{jn} \leq S^M \forall s \in S_{FIS}^n, n$$

(5)

4.1.4. Duration Constraints. We define $T_{jn}^d$ and $T_{jn}^l$ as the start and end times of unit $j$ at event point $n$, respectively. Then, the duration of event point $n$ on unit $j$ must exceed the processing time, as indicated by eq 6.

$$\begin{equation}
T_{jn}^d \geq T_{jn}^l + \sum_i \sum_{n \leq s \leq s + \Delta n} (\alpha_{i,j} w_{ij} + \beta_{i,j} b_{ij}) \forall j, n
\end{equation}$$

(6)

4.1.5. Sequencing Constraints for Different Tasks in the Same Unit. Event point $(n + 1)$ on unit $j$ must start after event point $n$ on this unit ends, as given by eq 7.

$$T_{jn}^s \geq T_{jn}^l \forall j, n \leq N$$

(7)

To correctly sequence different tasks in different units, we define new continuous variables $T_{jn}$ to denote the time when state $s$ produced in unit $j$ is available at event point $n$. The time when state $s$ is available at $(n + 1)$ is always after the time when it is available at $n$.

$$T_{jn} \leq T_{jn+1} \forall s \in S^n, j \in J^n, n \leq N - 1$$

(8)

where set $J^n$ denotes the processing units that can process tasks to produce state $s$.

If state $s$ is produced in unit $j$ at event point $n$, then the time when this state $s$ is available at event point $n$ should always be after the end time of this event point $n$ on unit $j$. In other words, this state $s$ is available for storage or consumption only after it is produced.

$$T_{jn} \geq T_{jn}^l - M \left[1 - \sum_i \sum_{n - \Delta n \leq s \leq n} w_{ij} \right] \forall s \in S^n, j \in J^n, n \leq N$$

(9)

If state $s$ is produced or temporarily held in unit $j$ at event point $n$, then the time when this state $s$ is available at $n$, must be before the start time of event point $(n + 1)$ on this unit.

$$T_{jn} \leq T_{jn}^l + M \left[1 - \sum_i \sum_{n - \Delta n \leq s \leq n} w_{ij} \right] \forall s \in S^n, j \in J^n, n \leq N$$

(10)

4.1.6. Material Transfer. Material transfer between processing units and storage in batch scheduling has been studied in the literature. Non-instantaneous material transfer, zero-wait material transfer, and non-zero transfer times are also implicitly or explicitly taken into account. However, the differentiation of these material transfers based on the availability of the storage is not explicitly discussed. In this work, we explicitly introduce the concept of indirect and direct material transfer, which allows us to differentiate material transfer between processing units and storage based on the availability of storage capacity and rigorously sequence and align tasks in different units. As shown in Figure 5, there are four scenarios of material transfer. First, materials can be transferred to storage immediately after production (e.g., material transfer MT1). In scenario 2, materials can be held in the processing units for some duration after production and then transferred to storage (e.g., MT2). If the storage capacity is large enough, materials produced or temporarily held in a unit can be first transferred to storage and then transferred to downstream units for further processing (e.g., MT3). This scenario is called indirect material transfer. If the storage capacity is not large enough, then some materials must be transferred directly to downstream processing units (e.g., MT4), which is direct material transfer. We generally classify material transfer as indirect and direct material transfer. Note that indirect and direct material transfer can take place simultaneously.

4.1.6.1. Indirect Material Transfer. In this scheme, the storage capacity is usually large enough. A state subject to UIS policy is always indirectly transferred to downstream processing units. For a state with FIS policy, indirect material transfer of the state works only when there is enough storage capacity. To model this indirect material transfer scheme, we define new binary variables $z_{jn}$.

$$z_{jn} = \begin{cases} 
1 & \text{if a state produced in unit } j \text{ at } (n - 1) \text{ is indirectly transferred to unit } j' \text{ at } n \\
0 & \text{otherwise}
\end{cases}$$

Figure 5. Indirect and direct material transfer.
If indirect material transfer happens from unit $j$ to unit $j'$ at event point $(n + 1)$, the start time of event point $(n + 1)$ on unit $j'$ should exceed the end time of event point $n$ on unit $j$.

$$T_{jn}^i \leq T_{j'n}^{i' + 1} + M \cdot [1 - z_{jj'}^{(n+1)}] \quad \forall \, j', \, j \in \text{CJ}_j \, j \neq j', \, n < N \quad (11)$$

where set $\text{CJ}_j$ denotes units consuming a state produced in unit $j$, that is, $\text{CJ}_j = \{j' \mid \exists \, s \in \mathbb{S}^j : j' \in \mathbb{F}_{j'}^s, j' \in \mathbb{F}_{j'}^s\}$, in which set $\mathbb{F}_{j'}^s$ denotes the processing units that can process tasks to consume state $s$.

We also define continuous variables $bT_{jj'}^{ij}$ to denote the amount of a state through indirect transfer between unit $j$ and unit $j'$. Specifically, the state is produced by a task in unit $j$ at event point $(n - 1)$ and then indirectly transferred to a downstream unit $j'$ in which task $i'$ takes place and consumes the transferred state at event point $n$. The total amount of state $s$ required by downstream tasks at event point $n$ should not exceed the amount stored at the previous event point $(n - 1)$ plus the amount through indirect transfer.

$$\sum_{j'} \sum_{i' \in (1, N_{r,j})} \left[ -\rho_{i'} \sum_{n \leq n' \leq n+\Delta_n} b_{ij'}^{(n',n)} \right] \leq ST_{j(n-1)}^\text{ml} + \sum_{j'} \sum_{i' \in (1, N_{r,j})} \sum_{j'' \in (1, N_{r,j})} bT_{jj''}^{ij} \quad \forall \, s \in \mathbb{S}^j, \, n > 1 \quad (12)$$

$$\sum_{j'} \sum_{i' \in (1, N_{r,j})} \left[ -\rho_{i'} \sum_{n \leq n' \leq n+\Delta_n} b_{ij'}^{(n',n)} \right] \leq ST_{j(n-1)}^0 + \sum_{j'} \sum_{i' \in (1, N_{r,j})} \sum_{j'' \in (1, N_{r,j})} bT_{jj''}^{ij} \quad \forall \, s \in (\mathbb{S}^j \cap \mathbb{S}^{j'}), \, n = 1 \quad (13)$$

where set $\mathbb{S}^{j'}$ denotes the states that are initially held in the processing units performing production tasks of the state, that is, $\mathbb{S}^{j'} = \left\{ s \mid \sum_{i' \in (1, N_{r,j})} b_{ij'} > 0 \right\}$.

The amount of state $s$ produced by task $i$ in unit $j$ through indirect material transfer at event point $n$ should not exceed the batch size produced by this task at the previous event point $(n - 1)$. When there is indirect material transfer from a unit $j$ to a downstream unit $j'$ at the first event point, the transferred material comes from the batch of the state that is initially held in unit $j$ (i.e., $bs_{0j} > 0$).

$$\rho_{i'} \sum_{n \leq n' \leq n+\Delta_n} b_{ij'}^{(n',n)} \geq \sum_{j'} bT_{jj'}^{ij} \quad \forall \, s \in \mathbb{S}^j, \, i, i' \in (1, N_{r,j}), \, n > 1 \quad (14)$$

$$\rho_{i'} \cdot bs_{0j} \geq \sum_{j'} bT_{jj'}^{ij} \quad \forall \, s \in \mathbb{S}^j, \, i, i' \in (1, N_{r,j}), \, n = 1, \, bs_{0j} > 0 \quad (15)$$

Similarly, the amount of state $s$ indirectly transferred to a downstream unit $j'$ where task $i'$ taking place at event point $n$ should not exceed the amount required by this task $i'$.

$$-\rho_{i'} \sum_{n \leq n' \leq n+\Delta_n} b_{ij'}^{(n',n)} \geq \sum_{j'} bT_{jj'}^{ij} \forall \, s \in \mathbb{S}^j, \, i' \in (1, N_{r,j}), \, n > 1 \quad (16)$$

When there is no indirect material transfer between two processing units, the amount through the indirect material transfer should be zero, which is ensured by eqs 18 and 19.

$$\sum_{i', j} bT_{jj'}^{ij} \leq \min[B_{ij}^{\text{max}}, B_{ij}^{\text{max}}] \cdot z_{jj'} \quad \forall \, j, j' \in \text{CJ}_j, \, j \neq j', \, n > 1 \quad (18)$$

$$\sum_{i', j} bT_{jj'}^{ij} \leq \min[B_{ij}^{\text{max}}, B_{ij}^{\text{max}}] \cdot z_{jj'} \quad \forall \, j, j' \in \text{CJ}_j, \, j \neq j', \, n = 1, \, s \in \mathbb{S}^j \cap \mathbb{S}^{j'}, \, \sum_{i' \in (1, N_{r,j'})} \sum_{i' \in (1, N_{r,j'})} b_{ij'} \geq 0 \quad (19)$$

where $B_{ij}^{\text{max}} = \max_{s, i \in (1, N_{r,j})} \{\rho_{i'} \cdot B_{ij}^{\text{max}}\}$ and $B_{ij}^{\text{max}} = \max_{s, i' \in (1, N_{r,j'})} \{\rho_{i'} \cdot B_{ij}^{\text{max}}\}$.

The abovementioned constraints are used to monitor indirect material transfer as well as the correct timing sequence between unit $j$ and unit $j'$ at event point $(n + 1)$. The indirect material transfer is not monitored from unit $j$ to unit $j'$ at event point $(n + 2)$ or higher event points. If a state produced in unit $j$ at event point $n$ is consumed by a task in unit $j'$ at event point $(n + 2)$ or higher event points, then the unit $j'$ that consumes state $s$ at event point $(n + 2)$ or higher event points should always start after the state is available on unit $j$ at event point $n$ to avoid storage violation. This logic condition is ensured by eqs 20 and 8.

$$T_{jn}^i \leq T_{j'n}^{i' + 2} + M \cdot \left[ 1 - \sum_{i' \in (1, N_{r,j'})} \sum_{n \leq n' \leq n+\Delta_n} w_{ij'}^{(n'+1)} \right] \quad \forall \, s \in \mathbb{S}^j, \, n < N - 1 \quad (20)$$

4.1.6.2. Direct Material Transfer. For a state subject to FIS policy, if there is no storage space, this state cannot be transferred to a dedicated storage tank. Instead, it must be transferred directly from unit $j$ to unit $j'$ where it is consumed.
To model direct material transfer, binary variables $z_{Djn}$ are introduced as follows:

$$z_{Djn} = \begin{cases} 
1 & \text{if a state produced in unit } j \text{ at } (n-1) \text{ is directly transferred to unit } j' \text{ at } n \\
0 & \text{otherwise} 
\end{cases}$$

If direct material transfer takes place from unit $j$ to $j'$ at $(n+1)$, unit $j'$ at event point $n$ must finish before unit $j$ finishes at the same $n$ to avoid overlapping on $j'$, as indicated in eq 21.

$$T_{jn}' \leq T_{jn} + M \cdot [1 - z_{Djn}(n+1)] \forall j, j' \in C_{jn}^F, j \neq j', n < N$$

where $C_{jn}^F = \{ j' | \exists s \in S_{FIS} \land j \in J_{s}^p, j' \in J_{s}^c \}$.

We define continuous variables $bTd_{j'n}$ to denote the amount of a state directly transferred from unit $j$ where it is produced by a task $i$ or temporarily held at event point $(n-1)$ to a downstream unit $j'$ where it is consumed by a task $i'$ at event point $n$. Direct material transfer of a state from its producing task $i$ on unit $j$ at event $(n-1)$ to the consuming task $i'$ on unit $j'$ at event point $n$ should be activated when there is no sufficient storage space to settle the amount of the state provided (produced or temporarily held in unit $j$) at event point $(n-1)$. Equation 22 indicates when the provided amount of one state plus the excess amount of the state at event point $(n-1)$ exceeds the maximum storage capacity, direct material transfer should be non-zero. Equation 23 enforces the inequality at the first event point.

$$\sum_{j} \sum_{i \in (I_j \cap I_{s}^c)} \rho_{ij} \cdot b_{j'n} \leq ST_{j'n}^n + \sum_{j'} \sum_{i \in (I_j \cap I_{s}^c)} \rho_{i'jn} \cdot b_{j'n}$$

$$+ \sum_{j} \sum_{i \in (I_j \cap I_{s}^c)} \sum_{i' \in (I_{i'} \cap I_{s}^c)} bTd_{j'n} \forall s \in S_{FIS} \land j, n > 1$$

$$\sum_{j} \sum_{i \in (I_j \cap I_{s}^c)} \rho_{ij} \cdot b_{j'n} \geq \sum_{j'} \sum_{i' \in (I_{i'} \cap I_{s}^c)} bTd_{j'n} + \rho_{i'jn} \cdot b_{j'n} \forall s \in S_{FIS} \land j, i \in (I_j \cap I_{s}^p), n > 1$$

Similar to indirect material transfer, eqs 24 and 25 enforce the amount of a state through direct material transfer and its remaining amount held in the processing unit at event point $n$ must not exceed the amount of this state provided by task $i$ on unit $j$ at the previous event point $(n-1)$.

$$\rho_{ij} \cdot b_{j'n} \leq b_{j'n}(n-1) + b_{s_j(n-1)}$$

$$\geq \sum_{j'} \sum_{i' \in (I_{i'} \cap I_{s}^c)} bTd_{j'n} + \rho_{i'jn} \cdot b_{j'n} \forall s \in S_{FIS} \land j, i \in (I_j \cap I_{s}^p), n > 1$$

Figure 6. Illustration of the meaning of eq 30 where the maximum storage capacity of state $S_1$ is 100 mu. (a) An infeasible schedule without eq 30 due to real-time storage violation; (b) a feasible solution generated with eq 30 (gray shading means the unit is processing a task not related to state $S_1$).
\[ p_s, b, 0, j \geq \sum_{j, i \in (I, \cap I^f)} b_{T, j, i, n} \forall s \in S^{FIS}, j \]
\[ i \in (I, \cap I^f), n = 1, b, 0, j > 0 \tag{25} \]

The amount of state \( s \) through direct material transfer to unit \( j' \) where task \( i \) takes place at event point \( n \) should not exceed the amount of this state consumed by task \( i' \) at event point \( n \).
\[ \rho_{s, j, i, n} \leq \sum_{j, i \in (I, \cap I^f)} b_{T, j, i, n} \forall s \in S^{FIS}, j, i \in (I, \cap I^f), n \]
\[ > 1 \tag{26} \]

\[ \rho_{s, j, i, n} \leq \sum_{j, i \in (I, \cap I^f)} b_{T, j, i, n} \forall s \in (S^{FIS} \cap S^{B}), j, i \in (I, \cap I^f), n \]
\[ \in (I, \cap I^f), n \]
\[ = 1 \tag{27} \]

When there is direct material transfer between two related processing units, the amount through this direct material transfer should be zero.
\[ \sum_{j, i \in (I, \cap I^f)} b_{T, j, i, n} \leq \min(B_{j}^{max}, B_{j}^{max}) - z_{D_j, n} \forall s \in S^{FIS}, j, \]
\[ j, i \in (I, \cap I^f), n \]
\[ > 1 \tag{28} \]

When the number of direct material transfer between two related processing units is limited, the amount through this direct material transfer should be zero.
\[ \sum_{j, i \in (I, \cap I^f)} b_{T, j, i, n} \leq \min(B_{j}^{max}, B_{j}^{max}) - z_{D_j, n} \forall s \in S^{FIS}, j, \]
\[ j, i \in (I, \cap I^f), n = 1, \]
\[ \sum_{s \in S^{FIS}, i \in (I, \cap I^f)} \sum_{i \in (I, \cap I^f)} b_{0, j} \]
\[ > 0 \tag{29} \]

4.1.7. Additional Sequence Constraints for FIS Policy. To avoid storage violation in real time, constraint 30 is also imposed to ensure that the time when state \( s \) produced in unit \( j \) is available at event point \( n \) must be after the time point of event point \( n - 1 \) on unit \( j' \) where a task consumes state \( s \) at event point \( n \). This is to ensure that when state \( s \) is available, there is enough space to hold this state, as illustrated in Figure 6. In Figure 6a, when \( S1 \) is available at the end of event point \( n \), there is no enough storage space for \( S1 \) because eq 30 is not imposed. However, if eq 30 is imposed, when \( S1 \) is available at the end of event point \( n \), it can be stored in the dedicated storage as \( S1 \) stored in the storage tank is transferred to unit \( J2 \) for processing, as shown in Figure 6b.

\[ T_{j, n} \geq T_{j' (n+1)} - M \cdot \left(1 - \sum_{j, i \in (I, \cap I^f)} \sum_{i \in (I, \cap I^f)} w_{j, i, n} \right) \forall s \in S^{FIS}, j \neq j', j \in J^f, j' \in J^c, j \neq j', n < N \tag{30} \]

An upper bound for the excess amount of state \( s \) stored is enforced by eq 31, which means the excess amount of state \( s \) stored at a time must not exceed the maximum storage capacity plus the maximum allowable holding amount in all processing units.
\[ ST_{j, n}^{M1} \leq ST_{j, n}^{max} + \sum_{i \in (I, \cap I^f)} \left[ \max_{i \in (I, \cap I^f)} \left( \rho_{s, j, i, n} \cdot B_{j, i}^{max} \right) \right] \forall s \in S^{FIS}, n \tag{31} \]

4.1.8. Allowing Processing Units to Store Materials. The amount of a state produced by task \( i \) that is temporarily held in unit \( j \) at event point \( n \) must be zero if the binary variable \( y_{s, j, n} = 0 \).
\[ b_{s, j, n} \leq B_{j}^{max} \cdot y_{s, j, n} \forall j, i \in (I, \cap I^f), n \tag{32} \]

The amount of a state held in a processing unit at the first event point must be smaller than the initial holding amount, as ascertained by eq 33.
\[ b_{s, j, n} \leq b_{0, j} \forall j, i \in (I, \cap I^f), n = 1 \tag{33} \]

4.1.9. Tightening Constraints. Some new tightening constraints are introduced to improve the performance of the proposed model. Constraints (34)–(37) relate \( w_{j, n} \) and \( y_{s, j, n} \) with \( z_{D_, n} \). Equation 34 ensures if unit \( j' \) processes a task consuming intermediate state \( s \), and indirect material transfer happens between units \( j \) and \( j' \) at event point \( n + 1 \), unit \( j \) must process a task producing \( s \) or temporarily hold \( s \) at event point \( n \).
\[ \sum_{j, i \in (I, \cap I^f)} \left( \sum_{i, n, \Delta n, s \leq n} w_{j, i, n} + y_{s, j, n} \right) \]
\[ \geq \sum_{j, i \in (I, \cap I^f)} \sum_{i, n, \Delta n, s \leq n + 1 + \Delta n} w_{j, i, n} + z_{D_, n} - 1 \forall \]
\[ s \in S^{in}, j \in J^f, j' \in J^c, j \neq j', j' \in J^c, n < N \tag{34} \]

Where set \( CJ0 \), includes units \( j' \in CJ \), except those in which multiple states produced in \( j \) are consumed by a task in \( j' \) or multiple states consumed by tasks in \( j' \) are produced by a task in \( j \).

Similarly, if unit \( j \) processes a task producing state \( s \) at event point \( n \), and there is indirect material transfer between units \( j \) and \( j' \) at event point \( n + 1 \), then unit \( j' \) must process a related consumption task starting at event point \( n + 1 \) according to constraint (35).
\[ \sum_{j, i \in (I, \cap I^f)} \sum_{i, n, \Delta n, s \leq n + 1 + \Delta n} w_{j, i, n} + y_{s, j, n} + z_{D_, n} - 1 \forall \]
\[ s \in S^{in}, j \in J^f, j' \in J^c, j \neq j', j' \in J^c, n < N \tag{35} \]

When direct material transfer between two units takes place (i.e., \( z_{D_, n} = 1 \)), indirect material transfer should also take place between these two units without loss of generality.
\[ z_{D_, n} \geq z_{D_, n} \forall j, j' \in J^f, j \neq j', n > 1 \tag{36} \]

When indirect material transfer takes place between units \( j \) and \( j' \) in set \( CJ \), at event point \( n \), unit \( j \) must process or hold a state at event point \( n - 1 \) and \( j' \) must process a task at event point \( n \).
2 - z_{I, j^n} \leq \sum_{i \in I} \sum_{n = 1 - \Delta n \leq n \leq 1} w_{j^n(n-1)} + \sum_{i \in I \setminus I^T} \sum_{n = 1 - \Delta n \leq n \leq 1} y_{j^n(n-1)} \\
+ \sum_{i \in I} \sum_{n = 1 - \Delta n \leq n \leq 1} w_{j^n(n)} \forall j, j' \in C, j \neq j' \\
\sum_{s \in (s^1 \cup s^2), (i, j, n) \in I} 1 = 0, n > 1 \tag{37}

4.1.10. Bounds and Fixing. The start and end times of unit j at event point n must be smaller than the time horizon (H), as ascertained by eqs 38 and 39. The parameter H would be replaced by M while addressing minimization of makespan, as the scheduling horizon (H) may be unavailable.

T_{jn}^s \leq H \forall j, n \tag{38}

T_{jn}^f \leq H \forall j, n \tag{39}

If all states produced by task i are subject to UIS policy, y_{s, j^n} and b_{s, j^n} should be zero.

y_{s, j^n} = 0 \forall i \notin I^s, j, n \tag{40}

b_{s, j^n} = 0 \forall i \notin I^b, j, n \tag{41}

While eq 42 imposes task i in unit j starting at event point n cannot finish at event point n' when n' < n, eq 43 ensures that a task must be processed in its suitable unit.

w_{j^n(n-1)} = 0, b_{j^n} = 0 \forall i, j, n, n' < n \tag{42}

w_{j^n(n)} = 0, b_{j^n} = 0 \forall i, j, i \notin I^f, n, n' \tag{43}

If there is no initial amount of a state held in a processing unit, the variables related to temporarily holding materials and material transfer are fixed as zero at the first event point.

b_{Ti, j'} = b_{T, j'} = z_{I, j'} = y_{s, j^n} = bs_{j^n} = 0 \forall i, j, j', n = 1, b_{0, j} = 0 \tag{44}

4.1.11. Objective Functions. As already discussed, two objective functions are considered. While constraint (45) is used for maximization of profit, constraint (46) is used for minimization of makespan.

z = \sum_{s \in S^1} \left[ p_s \sum_{j \in I} \sum_{I, n} \sum_{n = 1 - \Delta n \leq n \leq 1} (\rho_{i, j} \cdot b_{j^n(n)}) \right] \tag{45}

T_{jn}^f \leq MS \forall j, n = N \tag{46}

In minimization of makespan, it should be also ensured that the total demand is satisfied.

ST_{0, j} + \sum_{j \in I} \sum_{I, n} \sum_{n = 1 - \Delta n \leq n \leq 1} (\rho_{i, j} \cdot b_{j^n(i)}) \geq D_j \forall s \in S^p \tag{47}

The tightening constraint 48 is added for minimization of makespan only.

\sum_{i \in I} \sum_{n = 1 - \Delta n \leq n \leq 1} (\alpha_{i, j^n} \cdot w_{j^n(n-1)} + \beta_{i, j^n} \cdot b_{j^n(n)}) \leq MS \forall j \tag{48}

Finally, (49) and (50) denote all the continuous and binary variables of the model.

b_{j^n(1)}, \ b_{j^n(0)}, \ b_{Ti, j'}, \ b_{T, j'}, \ MS, \ ST_{0, j}, \ T_{jn}^s, \ T_{jn}^f \geq 0 \tag{49}

w_{j^n(1)}, \ y_{s, j^n}, \ z_{D, j^n}, \ z_{I, j^n} \in \{0, 1\} \tag{50}

We complete the mathematical model M1, which consists of constraints (1–45) and (49, 50) for maximization of profit, and (1–44) and (46–50) for minimization of makespan.

4.2. Model M2. In this model M2, the amount stored in dedicated storage and processing units is explicitly separated, which is different from model M1. As many constraints in this model are the same as those in model M1, the constraints that are different from model M1 are discussed in detail below.

We define new variables ST_{0, j} to denote the amount of state s stored in its dedicated storage tank at event point n. This amount denoted by ST_{0, j} is the amount denoted by ST_{0, j} in model M1 minus the amount stored in suitable processing units. Then, the material balance changes to the following eqs S1 and S2 using ST_{0, j}.

ST_{0, j}^f = ST_{0, j}^s(1) + \sum_{j \in I} \sum_{I, n} \sum_{n = 1 - \Delta n \leq n \leq 1} \rho_{i, j} \cdot b_{j^n(1)} \\
+ \sum_{j \in I} \sum_{I, n} \sum_{n = 1 - \Delta n \leq n \leq 1} \rho_{i, j} \cdot b_{j^n(0)} \tag{51}

ST_{0, j} = ST_{0, j} + \sum_{j \in I} \sum_{I, n} \sum_{n = 1 - \Delta n \leq n \leq 1} \rho_{i, j} \cdot b_{j^n(i)} \\
+ \sum_{j \in I} \sum_{I, n} \sum_{n = 1 - \Delta n \leq n \leq 1} \rho_{i, j} \cdot b_{j^n(i)} \\
- \sum_{j \in I} \sum_{I, n} \rho_{i, j} \cdot b_{j^n} \forall s, n \tag{52}

> 1

Similar to eq 12, the total amount of states required by downstream tasks at event point n should not exceed the amount stored at the previous event point (i.e., ST_{0, j}^s(1)) plus the amount through indirect transfer.

\sum_{j' \in I^f} \sum_{I, n} \sum_{n = 1 - \Delta n \leq n \leq 1} \left( -\rho_{i, j'} \cdot b_{j'^n(i)} \right) \leq ST_{0, j}^s(1) + \sum_{j} \sum_{I, n} \sum_{n = 1 - \Delta n \leq n \leq 1} b_{Ti, j'^n} \forall s \in S^p, n \tag{53}

The amount of state s through indirect material transfer at event point n should not exceed its available amount produced from task i or held in the processing unit at event point (n - 1).
\[ T_{m}^{M2} \geq T_{j(n+1)}^{f} - M \left( 1 - \sum_{i \in (1, N_{T'2})} \sum_{n=\Delta S_{1}}^{n_{2}} w_{j(n+1)}^{m,n} \right) \quad \forall \]
\[ s \in S_{m}^{in}, j \in J_{s}^{B}, n \]

Similar to constraint (20), if a state produced in unit at event point \( n \) is consumed by a task in unit at event point \( (n+2) \) or higher event points, the unit \( j' \) that consumes state \( s \) at \((n+2)\) or higher event points should always start after the state is available on unit at \( n \) to avoid storage violation. This logic condition is ensured by eqs 61 and 59.

\[ T_{m}^{M2} \leq T_{j(n+2)}^{f} + M \cdot \left[ 1 - \sum_{i \in (1, N_{T'2})} \sum_{n=\Delta S_{1}+1}^{n_{2}} w_{j(n+2)}^{m,n} \right] \quad \forall \]
\[ s \in S_{m}^{in}, j' \in J_{s}^{C}, n \]
\[ < N - 1 \]

Equations 62–64 are used to ensure that a unit \( j \) that processes or temporarily holds one batch of a production task for state \( s \) at event point \( n \) should finish after the end time of unit \( j' \) at the previous event point \((n-1)\), where a task consumes the state \( s \) at event point \( n \), to avoid real-time storage violation. Here, \( t_{m}^{M2} \) are intermediate variables. Handled by eqs 62 and 63, the unit \( j' \) consumes or pre-processes a state \( s \) before its production in unit \( j \) at the same event point to avoid the inventory level of state \( s \) after production in units at \( n \) beyond the maximum storage capacity.

\[ h_{m}^{M2} \leq T_{j(n+1)}^{f} + M \cdot \left[ 1 - \sum_{i \in (1, N_{T'2})} \sum_{n=\Delta S_{1}}^{n_{2}} w_{j(n+1)}^{m,n} + y_{j(n+1)}^{m,n} \right] \quad \forall \]
\[ s \in S_{m}^{FIS}, j \in J_{s}^{B}, n \]
\[ < N \]

\[ h_{m}^{M2} \geq T_{j(n+1)}^{f} - M \cdot \left[ 1 - \sum_{i \in (1, N_{T'2})} \sum_{n=\Delta S_{1}}^{n_{2}} w_{j(n+1)}^{m,n} \right] \quad \forall \]
\[ s \in S_{m}^{FIS}, j' \in J_{s}^{B}, n > 1, n < N \]

\[ h_{m}^{M2} \geq h_{m}^{M2} \quad \forall \ s \in S_{m}^{FIS}, n < N - 1 \]

The available time of any state at the last event point should be smaller than makespan while addressing the minimization of makespan.

\[ T_{m}^{min} \leq MS \quad s \in S_{m}^{in}, n = N \]

Some upper bounds are also imposed to this model 2 as better computational performance is obtained based on our computational experience. Batch size of tasks and amount for one batch of materials held in a suitable processing unit should be always smaller than the maximum unit capacity, as handled by eqs 66 and 67.

\[ b_{j(n+1)}^{m,n} \leq b_{j(n+1)}^{max} \quad j, i \in I_{j}, n, n \leq n' \leq n + \Delta n \quad \forall \]
\[ b_{j(n+1)}^{m,n} \leq b_{j(n+1)}^{max} \quad j, i \in I_{j}, n \]
Equations 68 and 69 express that the amount of state $s$ through indirect or direct material transfer cannot exceed its maximum production amount from a task $i$ at unit $j$ and the maximum consumption amount by a task $i'$ at unit $j'$.

$$b_{Ti_{ij'}} \leq \min\{[\rho_{ti}b_{ij}^{max}], (\rho_{ti}b_{ij}^{max})\} \forall s \in S^{TIS}, j, i \in (I_j \cap I_{j'}), i', j' \in (I_j \cap I_{j'}'), n$$

(68)

$$b_{Td_{ij'}} \leq \min\{[\rho_{ti}b_{ij}^{max}], (\rho_{ti}b_{ij}^{max})\} \forall s \in S^{in}, j, i \in (I_j \cap I_{j'}), i', j' \in (I_j \cap I_{j'}'), n$$

(69)

All continuous variables of model $M2$ are denoted by eq 70.

$$b_{v_{ij}}, b_{s_{ij}}, b_{Ti_{ij'}} b_{Td_{ij'}}, M, S^{TIS}, T_j^{M2}, T_i^{M2}, T_s^{I}, T_{\rho}^{M}, n \geq 0$$

(70)

The mathematical model $M2$ consists of constraints (1, 2), (6, 7), (11), (13), (16–19), (21), (24–29), (32–45), (50–68), and (70) for maximization of profit, and (1, 2), (6, 7), (11), (13), (16–19), (21), (32–45), (32–44), (46–48), and (50–70) for minimization of makespan. The complete model $M2$ is provided in Section S7 of the Supporting Information.

4.3. Extensions. The proposed model $M1$ can be extended to address the following situations like variable recipes, and states subject to NIS and ZW policies. We use model $M1$ for this extension due to its better computational performance than $M2$ (see Section 5).

4.3.1. Variable Recipes. The proposed model can be extended to handle schedule problems with variable recipe of a task, which indicates flexible proportions of output or input materials are used in a task. Such features have been addressed in the literature for some industrial applications. We define two new continuous variables $b_{v_{ij}}$ and $b_{s_{ij}}$, to denote the amount of state $s$ produced by task $i$ in unit $j$ from $n'$ to $n$ and the amount of state $s$ in a batch produced by task $i$ and held in unit $j$ at $n$, respectively. Set $I'$ indicates tasks with variable recipes. The amount of state $s$ produced by task $i \in I'$ is bounded by maximum and minimum fractions.

$$b_{v_{ij}}\rho_{d_i}^{min} \leq b_{v_{ij}} \leq b_{v_{ij}}\rho_{d_i}^{max} \forall s, j, i \in (I' \cap I_j), n, n' \leq n + \Delta n$$

(71)

The total amount of states produced by task $i \in I'$ should equal the batch size of the task.

$$\sum_{n \in I'} b_{v_{ij}} = b_{v_{ij}} \forall j, i \in (I' \cap I_j), n, n \leq n' \leq n + \Delta n$$

(72)

Amounts of one state in one batch, produced by task $i$ at previous events and temporarily held in the processing unit $j$ at event $n$, are bounded by the maximum and minimum fractions.

$$b_{s_{ij}}\rho_{d_i}^{min} \leq b_{s_{ij}} \leq b_{s_{ij}}\rho_{d_i}^{max} \forall s, j, i \in (I' \cap I_j \cap I''), n = 1, b_{s_{ij}} > 0$$

(73)

$$b_{s_{ij}}\rho_{d_i}^{min} \leq b_{s_{ij}} \leq b_{s_{ij}}\rho_{d_i}^{max} \forall s, j, i \in (I' \cap I_j \cap I''), n = 1, b_{s_{ij}} > 0$$

(74)

The amount in a batch held in unit $j$ at event point $n$ is calculated by eqs 75 and 76.

$$\sum_{i \in I'} b_{s_{ij}} = b_{s_{ij}} \forall j, i \in (I' \cap I_j \cap I''), n > 1$$

(75)

$$\sum_{i \in I'} b_{s_{ij}} = b_{s_{ij}} \forall j, i \in (I' \cap I_j \cap I''), n = 1, b_{s_{ij}} > 0$$

(76)

For task $i \in I'$, the amount of a state in a batch produced by task $i$ and temporarily held in unit $j$ at event point $(n + 1)$ is no more than its amount provided by the task $i$ at $n$.

$$b_{s_{ij}} + \sum_{n \in I'} b_{s_{ij}} \leq b_{s_{ij}} \forall s, j, i \in (I' \cap I_j \cap I' \cap I''), n$$

(77)

To handle variable production recipes for some tasks, constraints concerning material balance and material transfer should be modified. Specifically, variables $b_{v_{ij}}$ and $b_{s_{ij}}$ for task $i \in I'$ should be replaced by $b_{v_{ij}}$ and $b_{s_{ij}}$, respectively.

4.3.2. NIS and ZW Policies. A state with NIS policy can be either temporarily held in the processing unit or directly transferred into downstream processing units after its production, as there is no storage tank.

$$b_{Ti_{ij'}} = 0 \forall s \in (S^{TIS} \cup S^{ZW}), j, j', i \in (I_j \cap I_{j'}), i' \in (I_j \cap I_{j'}')$$

(78)

$$z_{ij'} = 0 \forall j, j', \sum_{i \in (S^{TIS} \cup S^{ZW})} \sum_{i' \in (I_j \cap I_{j'})} \rho_{d_i} = 0, n$$

(79)

Equations 78 and 79 are also true for a state with ZW policy as the state must be transferred to and processed immediately in the downstream processing units after production.

$$b_{s_{ij}} = b_{s_{ij}} = b_{s_{ij}} = 0 \forall s \in S^{ZW}, j, i \in (I_j \cap I_{j'}), n$$

(80)

When task $i$ in unit $j$ produces a state subject to ZW policy at event point $n$, the duration of this event point $n$ over unit $j$ must be exactly equal to the processing time of this task $i$ at event point $n$, which is ensured by eqs 6 and 81. Sets $I^{ZW}$ and $I^{Pzw}$ denote tasks in corresponding units that are able to produce a state with ZW policy, respectively.

$$T_{\rho} = T_{\rho} + \sum_{i \in I^{ZW}} \sum_{n \in I^{ZW}} \rho_{d_i} + b_{v_{ij}}$$

(81)

Equation 82 enforces the end time of the producing tasks is earlier than the start time of the consuming tasks for direct transfer of a state subject to NIS or ZW policy.

$$T_{\rho} \leq T_{\rho}^{Pzw} + M [1 - z_{D_{ij'}}(n+1)]$$

(82)

$$j, j' \in (C_{i}^{ZW} \cup C_{j}^{ZW}), j \neq j', n < N$$

(83)
where $C_{j}^{\text{NIS}} = \{ j' \mid \exists s \in S_{j}^{\text{NIS}}: j \in J_{s}^{p}, j' \in J_{s}^{c} \}$ and $C_{j}^{\text{ZW}} = \{ j' \mid \exists s \in S_{j}^{\text{ZW}}: j \in J_{s}^{p}, j' \in J_{s}^{c} \}$.

For a state with ZW policy, a task consuming this state at event point $(n + 1)$ should immediately start after this state is produced at $n$, as pre-processing unit wait is not permitted.

$$T_{j'}^{e} \leq T_{j}^{f} + M \cdot \left[ 2 - zD_{j'}^{f(n+1)} \right]$$

$$- \sum_{i \in (I_{n}^{f} \cap I_{n}^{c})} w_{ij}^{n} \sum_{n \leq n' \leq n + \Delta n} b_{ij'}^{n(n-1)} + b_{ij}^{n(n-1)}$$

$$\forall s \in S_{j}^{\text{ZW}}, j \neq j', n < N$$

(83)

Equation 85 expresses that for a state subject to NIS or ZW policy, materials produced by a task or held at event point $(n - 1)$ should be consumed, or partially held in the processing units at event point $n$. Given that a state for ZW cannot be held in processing units, eq 86 is valid only if the state subject to NIS is initially stored at one processing unit.

$$\rho_{i} \left[ \sum_{n \leq n' \leq n + \Delta n} b_{i}^{n(n-1)} + b_{i}^{n(n-1)} \right]$$

$$= \sum_{j \in (I_{n}^{f} \cap I_{n}^{c})} b_{T'}^{j}(n) + \rho_{i}^{p} b_{i}^{n} \forall s \in (S_{j}^{\text{ZW}} \cup S_{j}^{\text{NIS}}), j$$

$$, i \in (I_{n}^{f} \cap I_{n}^{c}), n > 1$$

(85)

For a state subject to NIS or ZW policy, the amount of this state consumed must be from direct material transfer.

$$-\rho_{i}^{p} \sum_{n \leq n' \leq n + \Delta n} b_{i}^{n(n-1)}$$

$$= \sum_{j \in (I_{n}^{f} \cap I_{n}^{c})} b_{T'}^{j}(n) \forall s \in (S_{j}^{\text{ZW}} \cup S_{j}^{\text{NIS}}), j'$$

$$, i' \in (I_{n}^{f} \cap I_{n}^{c}), n > 1$$

(87)

For a state subject to ZW policy, excess amounts of this state must be zero.

$$ST_{m}^{\text{M1}} = 0 \forall s \in S_{j}^{\text{ZW}}, n$$

(89)

Equations 91 and 92 enforce that a task producing states subject to ZW cannot occur at the last event.

$$\sum_{j \in (I_{n}^{f} \cap I_{n}^{c})} \sum_{n \leq n' \leq n + \Delta n} w_{ij}^{n} = 0 \forall s \in S_{j}^{\text{ZW}}, n = N$$

(91)

$$\sum_{j \in (I_{n}^{f} \cap I_{n}^{c})} \sum_{n \leq n' \leq n + \Delta n} b_{ij}^{n(n-1)} = 0 \forall s \in S_{j}^{\text{ZW}}, n = N$$

(92)

Equation 93 expresses that a state subject to ZW or NIS policy must be directly transferred to a downstream processing unit or partially held in the processing unit $j$ for NIS after production. Similarly, materials used by consuming tasks at event $n$ must be transferred directly from production tasks finished at the previous event $(n - 1)$, given by eq 94.

$$\sum_{j \in (I_{n}^{f} \cap I_{n}^{c})} \sum_{n \leq n' \leq n + \Delta n} w_{ij}^{n} \rho_{i}^{p} + y_{ij}^{n}$$

$$\geq \sum_{n \leq n' \leq n + \Delta n} w_{ij}^{n(n-1)} + y_{ij}^{n} \forall s \in (S_{j}^{\text{ZW}} \cup S_{j}^{\text{NIS}}), j, i \in (I_{n}^{f} \cap I_{n}^{c}), n > 1$$

(93)

$$\sum_{j \in (I_{n}^{f} \cap I_{n}^{c})} \sum_{n \leq n' \leq n + \Delta n} w_{ij}^{n}$$

$$\geq 0 \forall s \in (S_{j}^{\text{ZW}} \cup S_{j}^{\text{NIS}}), j', i' \in (I_{n}^{f} \cap I_{n}^{c}), n > 1$$

(94)

The complete model with all extensions is denoted as EM1, which is provided in Section S9 of the Supporting Information. Note that models M1–M2 are developed for scheduling multipurpose batch plants for FIS and UIS, while the extended model EM1 can solve problems with variable proportion of states produced/consumed and four types of storage policies (e.g., FIS, NIS, UIS, and ZW). It should also be noted that although many sets are defined, most of them can be automatically determined based on sets $S_{i}^{R}, S_{i}^{S}, S_{i}^{P} \cup S_{i}^{FIS}, S_{i}^{UIS}, S_{i}^{NIS}, S_{i}^{ZW}, I, I_{n}$, and the parameter $\rho_{i}^{p}$.
5. COMPUTATIONAL STUDIES

To evaluate the performance of the proposed mathematical models M1−M2, we solve 16 examples including three motivating examples, 11 benchmark examples from the literature, a variant of benchmark example 3 (denoted as example 12), and the Kallrath example. Relevant data, STN representations, and values of $M$ in big-M constraints for all examples are provided in the Supporting Information. It should be noted that we only compare the performance of our models with model V&S when we solve the benchmarking examples and the Kallrath example, as V&S usually outperforms other continuous-time models. Although the multiple-grid slot-based model from Mostafaei and Harjunkoski eliminates some big-M constraints, it is not suitable for limited storage capacities and often leads to much greater number of event points. Detailed comparison of our models with Shaik and Floudas, Li and Floudas, Susarla, Li, and Karimi, and Mostafaei and Harjunkoski will be presented in our next contribution.

All models are run using the CPLEX 12.6.3/GAMS 24.6.1 on a desktop computer with AMD Ryzen 9 3900X 3.8 GHz and 48 GB RAM running Windows 10. The maximum computational time is set as 1 h for the motivating examples and benchmark examples, while it is set as 40,000 s for the industrial-scale Kallrath example. The relative optimality gap for all examples is set as zero.

### Table 1. Computational Results for Motivating Examples

| model | event points/H | RMILP | MILP | CPU time (s) | binary variables | continuous variables | constraints |
|-------|---------------|-------|------|-------------|------------------|----------------------|-------------|
| Motivating Example 1 (H = 8 h) | | | | | | | |
| V&S | 3 | 500.00 | 300.00 | 0.05 | 12 | 32 | 66 |
| L&F | 3 | 500.00 | 300.00 | 0.13 | 6 | 29 | 41 |
| SLK2 | 4 | 500.00 | 300.00 | 0.05 | 12 | 81 | 94 |
| M1 | 3 | 500.00 | 500.00 | 0.05 | 12 | 37 | 72 |
| M2 | 3 | 500.00 | 500.00 | 0.05 | 12 | 40 | 72 |

| Motivating Example 2 (Dₜ = 100 μs) | | | | | | | |
| V&S | 3 | 5.00 | 11.5 | 0.02 | 12 | 32 | 68 |
| L&F | 3 | 5.00 | 11.5 | 0.02 | 6 | 29 | 43 |
| SLK2 | 4 | 5.00 | 11.5 | 0.02 | 12 | 83 | 99 |
| VMM | 100 | 5.00 | 11.5 | 0.23 | 404 | 1006 | 2822 |
| M1 | 3 | 8.00 | 8.00 | 0.05 | 12 | 37 | 76 |
| M2 | 3 | 8.00 | 8.00 | 0.05 | 12 | 41 | 81 |

| Motivating Example 3 (Dₜ = 100 μs, Dᵣ = 200 μs) | | | | | | | |
| V&S | 10 | 27.00 | 27.00 | 0.05 | 114 | 350 | 898 |
| 18 | 19.00 | 25.00 | 44.5 | 210 | 638 | 1666 |
| M1 | 14 | 24.24 | 27.88 | 833.2 | 122 | 319 | 667 |
| 15 | 24.24 | 27.88 | 360 | 131 | 319 | 667 |
| M2 | 14 | 24.24 | 27.88 | 597 | 122 | 359 | 868 |
| 15 | 24.24 | 27.88 | 2578 | 131 | 385 | 933 |

### Table 2. Computational Results for Benchmark Examples for Minimization of Makespan (UIS)

| model | event points | RMILP | MILP (h) | CPU time (s) | binary variables | continuous variables | constraints |
|-------|--------------|-------|----------|-------------|------------------|----------------------|-------------|
| Example 1a (Dₜ = 2000 μs) | | | | | | | |
| V&S | 14 | 24.24 | 27.88 | 833.2 | 122 | 319 | 667 |
| 15 | 24.24 | 27.88 | 360 | 131 | 342 | 716 |
| M1 | 14 | 24.24 | 27.88 | 133.7 | 122 | 359 | 868 |
| 15 | 24.24 | 27.88 | 2578 | 131 | 385 | 933 |
| M2 | 14 | 24.24 | 27.88 | 165.3 | 122 | 348 | 812 |
| 15 | 24.24 | 27.88 | 3600 | 131 | 373 | 872 |

| Example 1b (Dₜ = 4000 μs) | | | | | | | |
| V&S | 23 | 48.47 | 52.07 | 1164 | 203 | 526 | 1108 |
| M1 | 23 | 48.47 | 52.07 | 597 | 203 | 593 | 1453 |
| M2 | 23 | 48.47 | 52.07 | 449.4 | 203 | 753 | 1352 |

a: Vooradi and Shaik; b: Li and Floudas; c: Susarla, Li, and Karimi; d: Velez, Merchan, and Maravelias. 
Event points for continuous-time models and H for VMM. Step size. δ = 0.5. Relative gap. ε = 3.33%. δ = 4.35%. Resource limit reached. Step size. δ = 1.
5.1. Motivating Examples. We revisit the motivating examples 1–3. The computational results are provided in Table 1. From Table 1, it can be observed that our models can lead to a 66.7% higher profit ($500) for motivating example 1 than other representative models like V&S, L&F, and SLK2 ($300) due to the added flexibility allowing materials produced to share storage burden. For motivating example 2, our models generate the optimal makespan 5.1. Motivating Examples. The computational results from solving 11 benchmark examples and a variant example of example 3 (i.e., example 12) are presented in Tables 2–5. The model statistics (e.g., the number of binary variables, continuous variables, and constraints) for V&S are different from those reported in Vooradi and Shaik as some binary variables are not fixed based on STN. If we fix them, we do get the same model statistics. We do not present results with tiny computational efforts in these tables. The complete results are provided in Section S3 of the Supporting Information.

Table 2 lists the results for makespan minimization with UIS. From Table 2, our models require significantly less computational time to solve examples 1a and 1b to optimality than V&S. Specifically, the computational time for example 1a is reduced by 84% (133.7 vs 833.2 s) using M1 with 14 event points compared with V&S. In addition, V&S requires more than 1 h to generate the optimal solution using 18 event points with Δn = 1 and 2. As a result, the total CPU time required to generate the optimal makespan of 19 h using our models is significantly decreased by 3 orders of magnitude according to the iterative procedure. This is mainly because V&S imposes if task \( i' \) produces state \( s \) at event point \( n \), it must finish after a task \( i'' \) starts to consume this state \( s \in S^\text{in} \) at \( n \) and before a task \( i''' \) starts to consume this state \( s \in S^\text{in} \) at \( (n + 2) \), regardless of whether task \( i' \) or \( i'' \) actually takes place at \( n \) or \( (n + 2) \), which is overcome by eqs 9, 10, 20, and 30 in M1 and eqs 59–64 in M2.

Table 3. Computational Results for Benchmark Examples for Maximization of Profit (UIS)\(^a\)

| model | event points | RMILP (s) | MILP ($) | CPU time (s) | binary variables | continuous variables | constraints |
|-------|-------------|----------|----------|--------------|-----------------|---------------------|-------------|
| V&S   | 9           | 6601.65  | 5038.05  | 12.8         | 77              | 204                 | 417         |
|       | 10          | 6601.65  | 5038.05  | 353.6        | 86              | 227                 | 466         |
| M1    | 9           | 6601.65  | 5038.05  | 4.7          | 77              | 229                 | 533         |
|       | 10          | 6601.65  | 5038.05  | 66.9         | 86              | 255                 | 598         |
| M2    | 9           | 6601.65  | 5038.05  | 5.3          | 77              | 223                 | 500         |
|       | 10          | 6601.65  | 5038.05  | 84.1         | 86              | 248                 | 560         |
| V&S   | 8           | 4291.68  | 3738.38  | 10.99        | 134             | 364                 | 772         |
|       | 9           | 4488.96  | 3738.38  | 736.0        | 152             | 411                 | 878         |
| M1    | 8           | 4291.68  | 3738.38  | 10.88        | 120             | 540                 | 876         |
|       | 9           | 4488.96  | 3738.38  | 417.9        | 136             | 613                 | 1000        |
| M2    | 8           | 4291.68  | 3738.38  | 10.56        | 120             | 334                 | 790         |
|       | 9           | 4488.96  | 3738.38  | 527.0        | 136             | 377                 | 899         |
| V&S   | 5           | 2100.00  | 1583.44  | 1.13         | 123             | 309                 | 682         |
|       | 6           | 2750.96  | 1583.44  | 721.1        | 151             | 374                 | 842         |
| M1    | 5           | 2100.00  | 1583.44  | 0.80         | 115             | 469                 | 815         |
|       | 6           | 2750.96  | 1583.44  | 42.0         | 141             | 576                 | 1016        |
| M2    | 5           | 2100.00  | 1583.44  | 0.83         | 115             | 296                 | 762         |
|       | 6           | 2750.96  | 1583.44  | 72.6         | 141             | 358                 | 943         |
| V&S   | 7           | 3369.69  | 2358.20  | 514.9        | 179             | 439                 | 1002        |
|       | 8           | 3618.64  | 2358.20  | 3600         | 207             | 504                 | 1162        |
| M1    | 7           | 3369.69  | 2358.20  | 70.1         | 167             | 683                 | 1217        |
|       | 8           | 3618.64  | 2358.20  | 2476         | 193             | 790                 | 1418        |
| M2    | 7           | 3369.69  | 2358.20  | 114.8        | 167             | 420                 | 1124        |
|       | 8           | 3618.64  | 2358.20  | 3600         | 193             | 482                 | 1305        |
| V&S   | 10          | 5225.86  | 4262.80  | 49.53        | 263             | 634                 | 1482        |
|       | 11          | 5644.59  | 4262.80  | 3600         | 291             | 699                 | 1642        |
| M1    | 10          | 5225.86  | 4262.80  | 42.25        | 245             | 1004                | 1820        |
|       | 11          | 5644.59  | 4262.80  | 2067         | 271             | 1111                | 2021        |
| M2    | 10          | 5225.86  | 4262.80  | 62.02        | 245             | 606                 | 1667        |
|       | 11          | 5644.59  | 4262.80  | 2125         | 271             | 668                 | 1848        |

“V&S: Vooradi and Shaik,” relative gap. \( ^a \)0.16%. \( ^b \)0.09%. \( ^c \)0.09%.
while solving example 1a with 15 event points, even though it has a relatively larger number of continuous variables and constraints.

Table 3 provides the computational results for maximization of profit with UIS. From Table 3, it is shown that the proposed models M1, M2, and V&S require the same number of event points to generate the optimal solution without allowing a task to span over multiple event points. However, our proposed models require a smaller number of binary variables for examples 2 and 3. This is because our models only examine if there is material transfer between processing units, while V&S tests if there is material transfer from a task to another. As multiple tasks share one processing unit in examples 2−3, our models lead to smaller model size and less computational time. Specifically, the computational time is reduced by 1 order of magnitude (721 vs 42 s from M1) using our models to find the best result of $1583.442 for Example 3a. Similar reduction in the computational time can also be observed in examples 3b and 3d. Although more constraints are involved in the proposed models, less computational effort is required due to additional tightening constraints imposed. Finally, M1 shows

Table 4. Computational Results for Benchmark Examples with Maximization of Profit (FIS)\(^a\)

| model | event points | RMILP | MILP ($) | CPU time (s) | binary variables | continuous variables | constraints |
|-------|--------------|-------|----------|-------------|-----------------|---------------------|-----------|
| V&S   | 7            | 3369.69 | 2358.20  | 950.7       | 383             | 553                 | 1794      |
|       | 8            | 3618.64 | 2358.20  | 3600        | 445             | 637                 | 2086      |
| M1    | 7            | 3369.69 | 2358.20  | 339.5       | 311             | 1097                | 1855      |
|       | 8            | 3618.64 | 2358.20  | 3600        | 361             | 1273                | 2164      |
| M2    | 7            | 3369.69 | 2358.20  | 420.5       | 311             | 630                 | 1789      |
|       | 8            | 3618.64 | 2358.20  | 3600        | 361             | 727                 | 2084      |
| V&S   | 10           | 5225.86 | 4262.80  | 98.31       | 569             | 805                 | 2670      |
|       | 11           | 5644.59 | 4262.80  | 3600        | 631             | 889                 | 2962.1    |
| M1    | 10           | 5225.86 | 4262.80  | 83.63       | 461             | 1625                | 2782      |
|       | 11           | 5644.59 | 4262.80  | 3600        | 511             | 1801                | 3091      |
| M2    | 10           | 5225.86 | 4262.80  | 69.23       | 461             | 921                 | 2674      |
|       | 11           | 5644.59 | 4262.80  | 3600        | 511             | 1018                | 2969      |
| V&S   | 5            | 300.00  | 180.00   | 0.17        | 97              | 160                 | 425       |
|       | 6 (Δn = 1)   | 300.00  | 210.00   | 0.16        | 117             | 180                 | 505       |
|       | 6 (Δn = 2)   | 360.00  | 180.00   | 0.28        | 120             | 194                 | 525       |
|       | 6 (Δn = 3)   | 360.00  | 400.00   | 0.16        | 132             | 197                 | 596       |
| M1    | 5            | 300.00  | 210.00   | 0.05        | 89              | 194                 | 509       |
| M2    | 5            | 300.00  | 210.00   | 0.06        | 89              | 196                 | 493       |
| V&S   | 6            | 400.00  | 200.13   | 0.38        | 84              | 149                 | 432       |
|       | 6 (Δn = 1)   | 400.00  | 200.13   | 1.27        | 104             | 169                 | 512       |
|       | 6 (Δn = 2)   | 400.00  | 300.00   | 0.38        | 120             | 185                 | 560       |
|       | 6 (Δn = 3)   | 400.00  | 400.00   | 0.16        | 132             | 197                 | 596       |
| M1    | 6            | 400.00  | 400.00   | 0.05        | 79              | 185                 | 536       |
| M2    | 6            | 400.00  | 400.00   | 0.14        | 79              | 209                 | 536       |
| V&S   | 10           | 400.00  | 200.13   | 10.81       | 148             | 253                 | 760       |
|       | 10 (Δn = 1)  | 400.00  | 200.13   | 173.9       | 184             | 289                 | 904       |
|       | 10 (Δn = 2)  | 400.00  | 200.13   | 485.2       | 216             | 321                 | 1000      |
|       | 10 (Δn = 3)  | 400.00  | 200.13   | 2020        | 244             | 349                 | 1084      |
|       | 10 (Δn = 4)  | 400.00  | 250.00   | 127.6       | 268             | 373                 | 1156      |
|       | 10 (Δn = 5)  | 400.00  | 300.00   | 113.6       | 288             | 393                 | 1216      |
|       | 10 (Δn = 6)  | 400.00  | 350.00   | 9.86        | 304             | 409                 | 1264      |
|       | 10 (Δn = 7)  | 400.00  | 400.00   | 1.77        | 316             | 421                 | 1300      |
| M1    | 10           | 400.00  | 400.00   | 0.16        | 139             | 317                 | 956       |
| M2    | 10           | 400.00  | 400.00   | 0.59        | 139             | 357                 | 956       |
| V&S   | 10           | 1795.48 | 989.03   | 2096        | 569             | 805                 | 2670      |
|       | 11           | 2208.35 | 989.03   | 3600        | 631             | 889                 | 2962.1    |
| M1    | 10           | 1795.48 | 1184.48  | 316.8       | 461             | 1625                | 2782      |
|       | 11           | 2208.35 | 1184.48  | 3600        | 511             | 1801                | 3091      |
| M2    | 10           | 1795.48 | 1184.48  | 189.9       | 461             | 921                 | 2674      |
|       | 11           | 2208.35 | 1201.39  | 3600        | 511             | 1018                | 2969      |

\(^{a}\)V&S: Vooradi and Shaik; relative gap: \(^{b}0.199\%\), \(^{c}0.15\%\), \(^{d}0.15\%\), \(^{e}1.41\%\), \(^{f}0.83\%\), \(^{g}3.11\%\), \(^{h}3.23\%\), \(^{i}5.74\%\), \(^{j}5.08\%\).
better performance than M2 as it can prove optimality for example 3b using 8 event points within 1 h, while M2 cannot.

Table 4 reports the computational results for benchmark examples to maximize profit subject to FIS policy. From Table 4, we can observe that the proposed models lead to a smaller model size and a reduction in the computational time by a factor of $2^{-3}$ for Example 3b. More importantly, our proposed model is proven to be more general as a better schedule with higher profit by 21.5% ($1201.39$ vs $989.03$) is generated for example 12 using our models compared to V&S. This is because the batch of materials is permitted to be temporarily held in the suitable processing units over consecutive event points in our models (see the schedule in Figure 7).

Specifically, one batch produced by separation at event point N6 is partially held in unit J4 from N7 to N10 to share the storage burden of states S6 and S7.

The results for examples 6, 8, and 9 listed in Table 4 prove that our proposed models do not require a task to span over multiple events (i.e., $\Delta n = 0$) due to allowing processing units to hold materials over multiple event points. Therefore, all proposed models lead to significantly smaller model size with less numbers of binary and continuous variables. For instance, our models require 40.2% ($79$ vs $132$) and 56.0% ($139$ vs $316$) fewer binary variables than V&S to generate the optimal solution for examples 8 and 9, respectively. As the iterative procedure is often conducted to estimate the appropriate event points, the total test time would be significantly reduced if tasks do not require to span over multiple events using our models.

Table 5 reports the computational statistics for makespan minimization with FIS. From Table 5, we can observe that M1 proves optimality using dramatically decreased computational time for example 1a and leads to a 16.9% ($2.06$ vs $2.48$%) reduction in the relative gap for example 1b, compared with V&S. For example 2b, all the proposed models M1–M2 can yield the optimal solution of 47.68 h without any task spanning over multiple event points, but V&S reaches the optimal solution with $\Delta n = 2$. Decrease in the number of binary variables by 40% ($648$ vs $1080$) and computational time by a factor of 43 ($82$ vs $3600$) are realized by M1 compared with V&S. Comparing with M2, M1 always shows better performances due to low relative gaps and short computational times.

| model | event points | RMILP (h) | MILP (h) | CPU time (s) | binary variables | continuous variables | constraints |
|-------|--------------|-----------|----------|--------------|------------------|----------------------|-------------|
| Example 3a ($D_{3a} = 2000$ mu) |
| V&S   | 14           | 24.24     | 27.88$^b$| 3600         | 226              | 371                  | 1031        |
| M1    | 14           | 24.24     | 27.88$^c$| 1969         | 213              | 450                  | 1243        |
| M2    | 14           | 24.24     | 27.88$^e$| 3600         | 213              | 465                  | 1210        |

Example 3b ($D_{3b} = 4000$ mu)

| V&S   | 24           | 48.47     | 52.07$^d$| 3600         | 396              | 641                  | 1801        |
| M1    | 24           | 48.47     | 52.07$^e$| 3600         | 373              | 780                  | 2183        |
| M2    | 24           | 48.47     | 52.07$^f$| 3600         | 373              | 805                  | 2120        |

Example 2b ($D_{3a} = 500$ mu, $D_{3b} = 400$ mu)

| V&S   | 21           | 47.46     | 47.74$^g$| 3600         | 768              | 1255                 | 3859        |
| M1    | 21 (\(\Delta n = 1\)) | 47.38     | 47.69$^h$| 3600         | 928              | 1415                 | 4499        |
| M2    | 21 (\(\Delta n = 2\)) | 47.38     | 47.68$^i$| 3600         | 1080             | 1567                 | 4955        |

Figure 7. Schedule for the example 12 using M1–M2 (profit = $1201.39$) (means one state waits in a unit before processing, means batch held in one unit after production).

V&S: Vooradi and Shaik,$^{17}$ relative gap. $^b0.08%$. $^c0.84%$. $^d2.48%$. $^e2.06%$. $^f3.06%$. $^g0.07%$. $^h0.65%$. $^i0.64%$. $^j0.07%$. $^k0.60%$. 
Table 6. Computational Results for Kallrath Examples with Minimization of Makespan

| model | event points/H | RMILP | MILP (h) | CPU time (s) | binary variables | continuous variables | constraints |
|-------|----------------|-------|----------|-------------|------------------|----------------------|-------------|
| V&S   | 10             | 16    | 32       | 10,761      | 888              | 5113                 |             |
| V&M   | 100            | 16    | 32       | 167         | 2448             | 11,093               |             |
| EM1   | 10             | 16    | 32       | 5277        | 591              | 4229                 |             |

Instance 1 \((D_{P_1} = D_{P_2} = 0, D_{P_3} = D_{P_4} = D_{P_5} = 20 \text{ mu})\)

Instance 2 \((D_{P_1} = D_{P_2} = 10 \text{ mu}, D_{P_3} = D_{P_4} = 20 \text{ mu}, D_{P_5} = 30 \text{ mu})\)

Instance 3 \((D_{P_1} = D_{P_2} = D_{P_3} = D_{P_4} = 18 \text{ mu})\)

Instance 4 \((D_{P_1} = D_{P_2} = 30 \text{ mu}, D_{P_3} = D_{P_4} = 20 \text{ mu}, D_{P_5} = 40 \text{ mu})\)

Instance 5 \((D_{P_1} = D_{P_2} = 60 \text{ mu}, D_{P_3} = D_{P_4} = 40 \text{ mu}, D_{P_5} = 80 \text{ mu})\)

Instance 6 \((D_{P_1} = D_{P_2} = 0, D_{P_3} = D_{P_4} = D_{P_5} = 200 \text{ mu})\)

Instance 7 \((D_{P_1} = D_{P_2} = 100 \text{ mu}, D_{P_3} = D_{P_4} = 200 \text{ mu}, D_{P_5} = 300 \text{ mu})\)

Instance 8 \((D_{P_1} = D_{P_2} = D_{P_3} = D_{P_4} = D_{P_5} = 180 \text{ mu})\)

Instance 9 \((D_{P_1} = D_{P_2} = 150 \text{ mu}, D_{P_3} = D_{P_4} = 200 \text{ mu}, D_{P_5} = 100 \text{ mu}, D_{P_6} = 200 \text{ mu})\)

Instance 10 \((D_{P_1} = D_{P_2} = 60 \text{ mu}, D_{P_3} = 80 \text{ mu}, D_{P_4} = 40 \text{ mu}, D_{P_5} = 80 \text{ mu})\)

**V&S**: Vorardi and Shaik; **V&M**: Velez, Merchant, and Maravelias; **V&E**: NA: no integer solution found after resource limit exceeded. Relative gap: \(\delta \%\), steps: \(\delta = 1.0, 0.5, 0.1\). Relative gap: \(\delta = 1.0, 0.5, 0.1\).

Specifically, Examples 1a and 2b are not solved to optimality by M2 within 1 h, while M1 finds the optimal solution in less than 2000 and 100 s, respectively. In addition, M1 reports a smaller relative gap for example 1b.

5.3. Kallrath Example. We solve the industrial-scale Kallrath example using the extended model EM1, V&S, and V&M with a resource limit of 40,000 s. The computational results are listed in Table 6. From Table 6, it can be observed that EM1 leads to 50% less computational time for instance 1 and better objective values for instances 2 and 4–10, compared to V&S. EM1 also requires less computational time (348 vs 447 s) to find the optimal solution for instance 3. VMM shows better computational performance, especially in computational efficiency, while solving instances 1–4 as all tasks have fixed and integer processing times in these instances. However, its dominance is diminished with the incremental customer demands. EM1 finds the best feasible solutions with optimality gaps smaller than 17% within 40,000 s compared with VMM and V&S for larger instances 6–9. For instance, a better solution with 18% (334 vs 408) lower makespan is generated using EM1 for instance 7 within the same time resource compared to VMM. This is mainly attributed to the huge number of time intervals required in VMM, leading to a dramatic rise in the model size and computational effort. For instance 10 with decimal processing times of tasks (see Table S30 in the Supporting Information), VMM is hard to find any feasible solution within 40,000 s using the greatest common divisor (i.e., \(\delta = 0.1\)) of processing times due to the excessive number of discrete variables required. EM1 leads to 7.6% (91.9 vs 99.6)
smaller makespan and over 45% (11.01 vs 20.23 and 25.5%) lower relative gap than V’S and VMM with $\delta = 0.5$.

To further illustrate the robustness and efficiency of the proposed models, we also solve all motivating and benchmark examples using CPLEX 12.8.0/GAMS 25.1.3. The computational results are provided in Section S3 of the Supporting Information. Similar conclusions can be made when using CPLEX 12.8.0.

6. CONCLUSIONS

In this work, we presented two novel unit-specific event-based MILP models for short-term scheduling of multipurpose batch plants. We introduced the concept of indirect and direct material transfer, allowing efficient sequence or alignment of different tasks in different units. A batch was allowed to be partially transferred to downstream facilities and held in processing time over multiple event points. A new continuous variable denoting the available time of one state was introduced to avoid the limitations in the existing unit-specific event-based models. New tightening constraints were proposed to reduce computational expense.

The computational results for the motivating examples and benchmark examples demonstrated that the proposed models require fewer numbers of discrete variables in most cases, especially where a processing unit can process multiple tasks, compared with the existing unit-specific event-based formulation. Interestingly, all tasks do not need to span over multiple event points to yield optimal solutions for all solved examples. As a result, the computational effort was significantly reduced by a factor ranging from 1 to 43 using the proposed models especially M1. In some cases, it was reduced by at least 1 order of magnitude. More importantly, the proposed models can generate the same or better objective values for all solved examples. The increase in profit can range from 21.5 to 67% of magnitude. More importantly, the proposed models can generate the same or better objective values for all solved examples.

As a result, the computational effort was significantly reduced by a factor ranging from 1 to 43 using the proposed models especially M1. In some cases, it was reduced by at least 1 order of magnitude. More importantly, the proposed models can generate the same or better objective values for all solved examples.

Unit-specific event-based time representation; STN representations and information of examples; computational results for examples; nomenclature; automatic determination of sets; model M1; model M2; model M3; extended model M1 (EM1); and differences between $T_{in}$ and $T_{ijn}$ (PDF).

AUTHOR INFORMATION

Corresponding Author

Jie Li – Centre for Process Integration, Department of Chemical Engineering, The University of Manchester, Manchester M13 9PL, U.K.; orcid.org/0000-0001-5196-2136; Phone: +44 (0) 161 306 8622; Email: jie.li-2@manchester.ac.uk; Fax: +44 (0) 161 236 7439

Authors

Dan Li – Centre for Process Integration, Department of Chemical Engineering, The University of Manchester, Manchester M13 9PL, U.K.; orcid.org/0000-0001-8754-0955

Nikolaos Rakovitis – Centre for Process Integration, Department of Chemical Engineering, The University of Manchester, Manchester M13 9PL, U.K.

Taicheng Zheng – Centre for Process Integration, Department of Chemical Engineering, The University of Manchester, Manchester M13 9PL, U.K.

Yueling Pan – Department of Chemical Engineering, The University of Manchester, Manchester M13 9PL, U.K.

Giorgos Kopanacos – Flexicon Limited, London ECIV IAZ, U.K.

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.iecr.2c01363

Author Contributions

D.L. and N.R. contributed equally to this work.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

D.L. and T.Z. appreciate financial support from the China Scholarship Council—The University of Manchester Joint Scholarship (201908130170 and 202106440020). N.R. acknowledges financial support from the postgraduate award by The University of Manchester. J.L. appreciates financial support from Engineering and Physical Sciences Research Council (EP/T03145X/1).

ABBREVIATIONS

STN state task network
RTN resource task network
FIS finite intermediate storage
ZW zero wait
UIS unlimited intermediate storage
NIS no intermediate storage

REFERENCES

(1) Kondili, E.; Pantelides, C. C.; Sargent, R. W. H. A general algorithm for short-term scheduling of batch operations-I. MILP formulation. Comput. Chem. Eng. 1993, 17, 211−227.

(2) Pantelides, C. C. Unified frameworks for optimal process planning and scheduling. Proceedings on the 2nd Conference on Foundations of Computer-Aided Operations, 1994; pp 253−274.
(3) Floudas, C. A.; Lin, X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review. Comput. Chem. Eng. 2004, 28, 2109−2129.

(4) Lee, H.; Maravelias, C. T. Discrete-time mixed-integer programming models for short-term scheduling in multipurpose environments. Comput. Chem. Eng. 2017, 107, 171−183.

(5) Zhang, X.; Sargent, R. W. H. The optimal operation of mixed production facilities-A general formulation and some approaches for the solution. Comput. Chem. Eng. 1996, 20, 897−904.

(6) Castro, P.; Barbosa-Pövoa, A. P. F. D.; Matos, H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. Ind. Eng. Chem. Res. 2001, 40, 2059−2068.

(7) Maravelias, C. T.; Grossmann, I. E. New General Continuous-Time State–Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants. Ind. Eng. Chem. Res. 2003, 42, 3056−3074.

(8) Sundaramoorthy, A.; Karimi, I. A. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. Chem. Eng. Sci. 2005, 60, 2679−2702.

(9) Li, J.; Karimi, I. A.; Sririnivasan, R. Recipe determination and scheduling of gasoline blending operations. AIChE J. 2010, 56, 441−465.

(10) Susarla, N.; Li, J.; Karimi, I. A. Novel approach to scheduling multipurpose batch plants using unit-slots. AIChE J. 2010, 56, 1859−1879.

(11) Li, J.; Karimi, I. A. Scheduling gasoline blending operations from recipe determination to shipping using unit slots. Ind. Eng. Chem. Res. 2011, 50, 9156−9174.

(12) Ierapetritou, M. G.; Floudas, C. A. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. Ind. Eng. Chem. Res. 1998, 37, 4341−4359.

(13) Shaik, M. A.; Floudas, C. A. Novel unified modeling approach for short-term scheduling. Ind. Eng. Chem. Res. 2009, 48, 2947−2964.

(14) Li, J.; Floudas, C. A. Optimal event point determination for short-term scheduling of multipurpose batch plants via unit-specific event-based continuous-time approaches. Ind. Eng. Chem. Res. 2010, 49, 7446−7469.

(15) Tang, Q. H.; Li, J.; Floudas, C. A.; Deng, M. X.; Yan, Y. B.; Xi, Z. M.; Chen, P. H.; Kong, J. Y. Optimization framework for process scheduling of operation-dependent automobile assembly lines. Optim. Lett. 2012, 6, 797−824.

(16) Seid, R.; Majzor, T. A robust mathematical formulation for multipurpose batch plants. Chem. Eng. Sci. 2012, 68, 36−53.

(17) Vooradi, R.; Shaik, M. A. Rigorous Unit-Specific Event-Based Model for Short-Term Scheduling of Batch Plants Using Conditional Sequencing and Unit-Wait Times. Ind. Eng. Chem. Res. 2013, 52, 12950−12972.

(18) Li, J.; Xiao, X.; Floudas, C. A. Integrated gasoline blending and order delivery operations: Part I. Short-term scheduling and global optimization for single and multi-period operations. AIChE J. 2016, 62, 2043−2070.

(19) Rakovitsis, N.; Zhang, N.; Li, J.; Zhang, L. P. A new approach for scheduling of multipurpose batch processes with unlimited intermediate storage policy. Front. Chem. Sci. Eng. 2019, 13, 784−802.

(20) Méndez, C. A.; Cerda, J. Optimal scheduling of a resource-constrained multiproduct batch plant supplying intermediates to nearby end-product facilities. Comput. Chem. Eng. 2000, 24, 369−376.

(21) Hui, C.; Gupta, A.; van der Meulen, H. A. A. A novel MILP formulation for short-term scheduling of multi-stage multi-product batch plants with sequence-dependent constraints. Comput. Chem. Eng. 2000, 24, 2705−2717.

(22) Méndez, C. A.; Cerda, J. An MILP continuous-time framework for short-term scheduling of multipurpose batch processes under different operation strategies. Optim. Eng. 2003, 4, 7−22.

(23) Harjunkoski, I.; Maravelias, C. T.; Bongers, P.; Castro, P. M.; Engell, S.; Grossmann, I. E.; Hooker, J.; Méndez, C.; Sand, G.; Wassick, J. Scope for industrial applications of production scheduling models and solution methods. Comput. Chem. Eng. 2014, 62, 161−193.

(24) Georgiadis, G. P.; Elekidis, A. P.; Georgiadis, M. C. Optimization-Based Scheduling for the Process Industries: From Theory to Real-Life Industrial Applications. Processes 2019, 7, 438.

(25) Vooradi, R.; Shaik, M. A. Improved three-index unit-specific event-based model for short-term scheduling of batch plants. Comput. Chem. Eng. 2012, 43, 148−172.

(26) Mostafaei, H.; Harjunkoski, I. Continuous-time scheduling formulation for multipurpose batch plants. AIChE J. 2020, 66, No. e16804.

(27) Shaik, M. A.; Vooradi, R. Short-term scheduling of batch plants: Reformulation for handling material transfer at the same event. Ind. Eng. Chem. Res. 2017, 56, 11175−11185.

(28) Lee, H.; Maravelias, C. T. Combining the advantages of discrete- and continuous-time scheduling models: Part I. Framework and mathematical formulations. Comput. Chem. Eng. 2018, 116, 176−190.

(29) Lee, H.; Maravelias, C. T. Combining the advantages of discrete- and continuous-time scheduling models: Part 3. General algorithm. Comput. Chem. Eng. 2020, 139, 106848.

(30) Hubbs, C. D.; Li, C.; Sahinidis, N. V.; Grossmann, I. E.; Wassick, J. M. A deep reinforcement learning approach for chemical production scheduling. Comput. Chem. Eng. 2020, 141, 106982.

(31) MowBray, M.; Zhang, D.; Del Rio-Chanona, E. A. Distributional reinforcement learning for scheduling of chemical production processes. 2022, arXiv:2203.00636.

(32) Velez, S.; Merchán, A. F.; Maravelias, C. T. On the solution of large-scale mixed integer programming scheduling models. Chem. Eng. Sci. 2015, 136, 139−157.

(33) Zynigter, D.; Kelly, J. D. Multi-Product Inventory Logistical Modeling in the Process Industries. Optimization and Logistics Challenges in the Enterprise; Springer: Boston, MA, 2009; pp 61−95.

(34) Prasad, P.; Maravelias, C. T.; Kelly, J. Optimization of aluminum smelter casthouse operations. Ind. Eng. Chem. Res. 2006, 45, 7603−7617.

(35) Giménez, D. M.; Henning, G. P.; Maravelias, C. T. A novel network-based continuous-time representation for process scheduling: Part I. Main concepts and mathematical formulation. Comput. Chem. Eng. 2009, 33, 1511−1528.

(36) Giménez, D. M.; Henning, G. P.; Maravelias, C. T. A novel network-based continuous-time representation for process scheduling: Part II. General framework. Comput. Chem. Eng. 2009, 33, 1644−1660.

(37) Castro, P.; Barbosa-Pövoa, A.; Matos, H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. Ind. Eng. Chem. Res. 2001, 40, 2059−2068.

(38) Ferrer-Nadal, S.; Capón-García, E.; Méndez, C. A.; Puigjaner, L. Material transfer operations in batch scheduling. A critical modeling issue. Ind. Eng. Chem. Res. 2008, 47, 7721−7732.

(39) Wu, O. V.; Ave, G.; Harjunkoski, I.; Bouaswaig, A.; Schneider, S. M.; Roth, M.; Imsland, L. Short-term multiproduct batch scheduling considering storage features. IFAC-PapersOnLine 2020, 53, 11794−11799.

(40) Castro, P. M.; Westerlund, J.; Forssell, S. Scheduling of a continuous plant with recycling of byproducts: A case study from a tissue paper mill. Comput. Chem. Eng. 2003, 33, 347−358.

(41) Blömer, F.; Günther, H. Scheduling of multiproduct batch process in the chemical industry. Comput. Ind. 1998, 36, 245−259.

(42) Westenberger, H.; Kallrath, L. Formulation of a Job Shop Problem in Process Industry, Internal Report; Bayer AG, BASF AG: Leverkusen, Ludwighafen, 1995.

(43) Kallrath, J. Planning and scheduling in the process industry. OR Spectr. 2002, 24, 219−250.