Research Article

Unit-Specific Event-Based and Slot-Based Hybrid Model Framework with Hierarchical Structure for Short-Term Scheduling

Yue Wang, Hongye Su, Hanshan Shao, and Lei Xie

State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China

Correspondence should be addressed to Hongye Su; hysu@iipc.zju.edu.cn

Received 27 March 2015; Accepted 26 July 2015

Academic Editor: Qing Chang

Copyright © 2015 Yue Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Unit-specific event-based continuous-time model has inaccurate calculation problems in involving resource constraints, due to the heterogeneous locations of the event points for different units. In order to address this limitation, a continuous-time unit-specific event-based and slot-based hybrid model framework with hierarchical structure is proposed in this work. A unit-specific event-based model without utility constraints is formulated in upper layer, and a slot-based model is introduced in lower layer. In the hierarchical structure, the two layers jointly address the short-term production scheduling problem of batch plants under utility consideration. The key features of this work include the following: (a) eliminating overstrict constraints on utility resources, (b) solving multiple counting problems, and (c) considering duration time of event points in calculating utility utilization level. The effectiveness and advantages of proposed model are illustrated through two benchmark examples from the literatures.

1. Introduction

Short-term production scheduling for batch process is extensively studied in the last two decades. Most literatures made considerable efforts to decrease the number of integer variables and model size to pursue better computational performance [1–3]. However, the resource consideration in production scheduling has not gained enough attention. Resources are closely related to production. In multipurpose batch plants, resource restrictions can affect product pathways and have a significant impact on plant profitability [4]. Moreover, not considering resources will lead to the infeasibility of production scheduling. Therefore, considering resource constraints in production scheduling is vital, especially for the shared resources. The resources consumed in production are usually classified into two types. One type is discrete resource (e.g., processing units, manpower, and tools) and the other type is continuous resource (e.g., utilities and materials).

Regarding discrete resource, some typical algorithms, such as rescheduling algorithm, relaxation method, and

particle swarm optimization algorithm, have been utilized successively in production scheduling with discrete resource constraints. Méndez and Cerdá [5, 6] firstly proposed a MILP formulation where all discrete resources including processing units are treated uniformly and then developed a new MILP reactive scheduling algorithm to revise the short-term schedule of the resource constrained multistage batch facilities due to unexpected events like equipment breakdowns and worker absenteeism. Pinto and Grossmann [7] introduced a novel relaxation method into continuous-time MILP scheduling models and efficiently modeled discrete resource constraints (e.g., number of operators and fixed amount of electricity). Zhang et al. [8] proposed a hybrid discrete particle swarm optimization algorithm to solve the dual-resource constrained job shop scheduling problem with resource flexibility. As results of these contributions, problems including processing unit and tool allocation and manpower changes were able to be solved effectively.

Utilities as one of the continuous resources are studied frequently in the literatures. Discrete-time representation

method is based on a uniform discretization of time, which makes it easier to handle utility constraints in a uniform model. Lindholm and Giselsson proposed a discrete-time MIQP model to address the utility disturbance in production process with the aim of minimizing the total economic loss of site [9]. In order to involve utility consideration in general and large-scale industrial sites, Lindholm and Nytzén [10] introduced a general framework for production scheduling (PS) and detailed production scheduling (DPS) using a two-level hierarchical approach. The PS and DPS are both represented by discrete-time models. Wang et al. [11] proposed a planning and scheduling model framework with hierarchical structure. In the scheduling layer, a variperiod discrete-time presentation model is formulated to address utility disturbance.

Although the discrete-time representation method can deal with utility constraints in production scheduling easily, it still has some distinct limitations compared with the continuous-time model, including the following: (i) it corresponds to an approximation of the time horizon and (ii) it may result in an unnecessary growth in the number of binary variables. Consequently, in the overall size of the model, it usually requires an exceptionally large number of integer variables for industrial applications [12]. To solve the inherent limitations of the discrete-time models, methods based on continuous-time representation have been developed. During the last two decades, numerous formulations have been proposed based on continuous-time representation, including slot-based, global-event-based, unit-specific event-based, and precedence-based continuous-time formulations. Continuous-time representation model has been used frequently to address utility constraints in production scheduling and has been studied by Behdani et al. [13], Susarla et al. [14], Shaik and Vooradi [15], Méndez and Cerdá [16], and so forth.

It is noteworthy that unit-specific event-based model is considered as the most general and rigorous representation of time used in short-term scheduling and also has received some attention in utility constraint problems. Janak et al. [12] (referred to as JLF model in this work) proposed an enhanced unit-specific event-based formulation for short-term scheduling of multipurpose batch processes. This model incorporated and elaborated several features including resource constraints. Shaik and Floudas [17] (referred to as S&F model in this work) introduced a novel unit-specific event-based short-term scheduling model with three-index binary and continuous variables, which efficiently merges the problems with and without resources into a unified, generic common framework. Vooradi and Shaik [18] (referred to as S&V model in this work) enhanced the three-index model of Shaik and Floudas [17] by effectively incorporating the concept of active task throughout the model, due to which the proposed formulation has fewer constraints and big- M terms.

Recently, inaccurate calculation problems of unit-specific event-based model in involving resource constraints are presented in some literatures [12, 17, 19]. Because of the heterogeneous locations of the event points for different units, the utility utilization levels for tasks performed in different

units at the same event points are difficult to be calculated. Moreover, the multiple counting of utility consumption frequently occurs, which leads to an unreal shortage of utility. These limitations result in the inaccuracy of utility record keeping.

In order to solve the above inherent limitations, in this work, we proposed a unit-specific event-based and fixed slot-based continuous-time hybrid model framework with hierarchical structure to address the inaccurate utility calculation problems. A unit-specific event-based continuous-time model is used to optimize the production schedule without utility consideration in upper layer, and the obtained solutions of the upper layer are regarded as the input of the lower layer model which is based on fixed slot-based time representation with utility constraints. A better scheduling solution is found compared with historical literatures through iterations between upper and lower layers.

2. Problem Statement

Continuous-time models are generally classified into slot-based, global-event-based, unit-specific event-based, and precedence-based formulations. In the slot-based models, the time horizon is divided into several unequal time intervals, and the tasks have to start and terminate at an event [20]. Global-event-based models use a set of events that are common across all units, and the event points are defined at either the beginning or end (or both) of each task in each unit [19]. Regarding the unit-specific event-based models, the unit refers to equipment (e.g., reactor) occupied by processing tasks in production. Event points are a sequence of time instances located along the time axis of a unit, and each event point represents the beginning of a task or the utilization of the unit. The location of the event points is different for each unit, allowing different tasks to start at different time in each unit for the same event point [12]. For sequential processes, other alternative approaches based on precedence relationships have also been developed which do not require the postulation of any events or time slots [17]. In this work, what we focus on is the formulations based on unit-specific event-based continuous-time representation.

Event-based continuous-time representation allows event points to be located in any timing in scheduling horizon. The start and finish times are allotted to each event point. In addition, unit-specific event-based continuous-time representation also allows the start and finish times of tasks in different units corresponding to the same event point to take place at different timing. As shown in Figure 1, at event point N , the start time T_s and finish time T_f of tasks i_1 , i_2 , and i_3 are all different. The unit-specific event-based model considers the time horizon in a “true” continuous-time form without requiring tiny time slots and requires relatively lower number of event points. Thus, unit-specific event-based continuous-time model is considered to be more realistic and precise compared with the global event-based continuous-time model.

However, for production scheduling involving shared resources, the unit-specific event-based continuous-time

FIGURE 1: Unit-specific event-based continuous-time representation.

model is defective. Some shared resources, for instance, utilities, are frequently consumed by different units at the same time. At each event point, total utility utilization levels should be calculated to meet the supply of utilities. But in the unit-specific event-based continuous-time representation, the start and finish times of different units at the same event point are inconsistent, which hinders the utilization of unit-specific event-based model in production scheduling involving shared resource constraints.

In involving shared resource constraints, primary limitations of unit-specific event-based continuous-time model are described as follows.

2.1. Overstrict Constraint Problem about Utility Resource. JLF [12], S&F [17], and V&S [18] models enforce the start time of tasks consuming the same utility to be equal to $T_{ut}^s(u, n)$ if the tasks are active at event n , where $T_{ut}^s(u, n)$ is start time at which there is a change in the consumption of utility u at event n . Constraints related to start and finish times of tasks consuming the same utility and start time of utility given in V&S [18] model are as follows:

$$T_{ut}^s(u, n) \geq T^s(i, n) - M \left(1 - \sum_{\substack{n' \in N \\ n - \Delta n \leq n' \leq n}} \sum_{\substack{n'' \in N \\ n \leq n'' \leq n' + \Delta n}} w(i, n', n'') \right), \quad \forall u \in U, i \in I_u, n \in N, \quad (1)$$

$$T_{ut}^s(u, n) \leq T^s(i, n) + M \left(1 - \sum_{\substack{n' \in N \\ n - \Delta n \leq n' \leq n}} \sum_{\substack{n'' \in N \\ n \leq n'' \leq n' + \Delta n}} w(i, n', n'') \right), \quad \forall u \in U, i \in I_u, n \in N.$$

In (1), M is a large positive number in big- M constraints. As shown in Figure 2, we assume that tasks i_{c1} , i_{c2} , and i_{c3} are performed in unit 1, unit 2, and unit 3, respectively, and tasks

FIGURE 2: Gantt chart for motivating example 1.

i_{c1} , i_{c2} , and i_{c3} all consume the same utility at event point N . Equation (1) can be rewritten as follows:

$$T_{ut}^s(u, n) \geq T^s(i_{c1}, n), \quad (2)$$

$$T_{ut}^s(u, n) \leq T^s(i_{c1}, n). \quad (3)$$

Combining (2) and (3) yields

$$T_{ut}^s(u, n) = T^s(i_{c1}, n). \quad (4)$$

Similarly,

$$T_{ut}^s(u, n) = T^s(i_{c1}, n) = T^s(i_{c2}, n) = T^s(i_{c3}, n). \quad (5)$$

The start times of i_{c1} , i_{c2} , and i_{c3} are enforced to be equal to $T_{ut}^s(u, n)$, which results in the fact that the constraints are too restrictive and the scheduling solution will not be optimal.

2.2. Multiple Counting Problem. The consumptions of utilities consumed by tasks which continue over several event points are multiple counted in JLF, S&F, and V&S models:

$$\sum_{i \in I_u} \left(\gamma_{iu} \sum_{n' \in N} \sum_{n'' \in N} w(i, n', n'') \right) + \sum_{i \in I_u} \left(\delta_{iu} \sum_{n' \in N} \sum_{n'' \in N} b(i, n', n'') \right) \leq U_u^{\max}, \quad (6)$$

$$\forall u \in U, n \in N, n - \Delta n \leq n' \leq n, n \leq n'' \leq n' + \Delta n.$$

As shown in Figure 3, we assume that task i_{c1} is performed in unit 1; tasks i_{c2} and i_{c3} are both performed in unit 2; task i_{c1} continues from events N to $N+1$; tasks i_{c2} and i_{c3} are active at N and $N+1$, respectively. Three tasks all consume the same utility. At events N and $N+1$, (6) from V&S model can be rewritten by

$$\begin{aligned} & \gamma_{i_{c2}u} \cdot w(i_{c2}, n, n) + \delta_{i_{c2}u} \cdot b(i_{c2}, n, n) + \gamma_{i_{c1}u} \\ & \cdot w(i_{c1}, n, n+1) + \delta_{i_{c1}u} \cdot b(i_{c1}, n, n+1) \leq U_u^{\max}, \\ & \gamma_{i_{c3}u} \cdot w(i_{c3}, n+1, n+1) + \delta_{i_{c3}u} \cdot b(i_{c3}, n+1, n+1) \\ & + \gamma_{i_{c1}u} \cdot w(i_{c1}, n, n+1) + \delta_{i_{c1}u} \\ & \cdot b(i_{c1}, n, n+1) \leq U_u^{\max}. \end{aligned} \quad (7)$$

FIGURE 3: Gantt chart for motivating example 2.

FIGURE 4: Gantt chart for motivating example 3.

As given in (7), the consumption of utility consumed by task i_{c1} from event point N to event point $N+1$ is calculated twice. This repetitive counting leads to an inaccurate calculation about the consumption of utility and results in an unreal shortage of utility.

2.3. Problem of Missing Duration Time of Event Points. In JLF, S&F, and V&S models, the consumption constraint about utility is considered but not involving the duration time of event points. As given by (6), the supply of utility is constant no matter how long the duration time of event points is. As shown in Figure 4, task i is active from event points $N-1$ to $N+1$ and consumes certain utility. Assume that the duration times of event points $N-1$, N , and $N+1$ are $2k$ hours, k hours, and $3k$ hours, respectively. The supply of utility at event points $N-1$, N , and $N+1$ is obviously different. Consequently, the supply of utility should be considered according to the duration time of event points.

Due to the heterogeneous locations of the event points for different units, utility constraints are difficult to be involved in unit-specific event-based continuous-time model in a uniform framework. In this work, a hierarchical model framework combining unit-specific event-based continuous-time presentation and fixed slot-based continuous-time presentation is proposed. As shown in Figure 5, the scheduling solution without utility consideration presented by unit-specific event-based method is mapped into a uniform time presentation, and the scheduling horizon is divided into several fixed time slots. By this time presentation method, utility constraints can be calculated easily and accurately.

3. Mathematical Formulation

Framework structure of upper and lower scheduling models proposed in this work is presented in Figure 6. In the upper layer, utility constraints are not involved. The objective of this layer is the maximization of profit. The calculated solutions of upper layer are allocated down to lower layer as assignments, including processing amount of materials, start time and finish time of tasks. The objective of lower layer is to complete

FIGURE 5: Uniform time presentation combining unit-specific event-based and slot-based method.

FIGURE 6: Framework structure of upper and lower scheduling models.

the scheduled processing amount of materials (including raw materials, intermediate products, and final products), within the scheduled time under utility constraints.

3.1. Upper Layer Model. Mathematical formulations in upper layer are based on the model proposed by Vooradi and Shaik [18], which allow tasks to continue over several event points, and incorporate the concept of active task. Parameter Δn is introduced to define the maximum number of event points over which a task is allowed to continue.

3.1.1. Allocation Constraint. Consider

$$\sum_{i \in I_j} \sum_{n' \in N} \sum_{\substack{n'' \in N \\ n - \Delta n \leq n' \leq n \leq n'' \leq n' + \Delta n}} w(i, n', n'') \leq 1, \quad (8)$$

$$\forall j \in J, n \in N.$$

In each unit, at most one task can be active at each event point as given by (8). If task i is active from event point n' to event point n'' , $w(i, n', n'') = 1$; otherwise $w(i, n', n'') = 0$.

3.1.2. Capacity Constraints. Consider

$$\begin{aligned} B_i^{\min} w(i, n, n') &\leq B^{(m)}(i, n, n') \\ &\leq [B_i^{\max} - \text{Diff}_{i,n,n'}^{(m)}] \cdot w(i, n, n'), \quad (9) \\ \forall i \in I, m \in M, n, n' \in N, n \leq n' \leq n + \Delta n. \end{aligned}$$

The capacity constraints express the fact that if task i starts at event point n and ends at event point n' , the amount of material processed by task i will be limited by the upper and lower bounds on the production capacity. If the task is not active, then these constraints enforce zero batch size. The parameter $\text{Diff}_{i,n,n'}^{(m)}$ is a feedback from lower layer in the m th iteration, and the formulations about $\text{Diff}_{i,n,n'}^{(m)}$ are given by (10) to (12), where m is iteration number.

If $m = 1$,

$$\text{Diff}_{i,n,n'}^{(m)} = 0, \quad i \in I, n, n' \in N, m \in M. \quad (10)$$

If $m = 2$, namely it is the first time that variable $\text{diffs}^{(m)}(i, n, n')$ is fed back,

$$\begin{aligned} \text{Diff}_{i,n,n'}^{(m)} &= B_i^{\max} - B^{(m-1)}(i, n, n') \\ &\quad + \text{diffs}^{(m-1)}(i, n, n'), \quad (11) \\ i \in I, n, n' \in N, m \in M, m = 2. \end{aligned}$$

If $m > 2$,

$$\begin{aligned} \text{Diff}_{i,n,n'}^{(m)} &= \text{Diff}_{i,n,n'}^{(m-1)} + \text{diffs}^{(m-1)}(i, n, n'), \quad (12) \\ i \in I, n, n' \in N, m \in M, m > 2. \end{aligned}$$

3.1.3. Material Balances. If $n = 1$, the amount of states (raw material or intermediate) at event point n is equal to the initial value and decreased by any amount consumed at event point n , as given by (13). The amount of final product at event point $n = 1$ is calculated as initial value, as given by (14):

$$ST(s, n) = ST_s^0 + \sum_{i \in I_s^c} \rho_{is}^c \sum_{\substack{n' \in N \\ n \leq n' \leq n + \Delta n}} B^{(m)}(i, n, n'), \quad (13)$$

$$\forall s \in S^R, S^I, n = 1, m \in M,$$

$$ST(s, n) = ST_s^0, \quad \forall s \in S^P, n = 1. \quad (14)$$

If $n > 1$, the amount of a state at current event point is adjusted by any amounts produced by tasks which ends at the previous event and by any amounts consumed by the tasks which starts at the current event, as given by

$$\begin{aligned} ST(s, n) &= ST(s, n-1) + \sum_{i \in I_s^c} \rho_{is}^c \\ &\quad \cdot \sum_{\substack{n' \in N \\ n \leq n' \leq n + \Delta n}} B^{(m)}(i, n, n'), \\ \forall s \in S^R, n \in N, n > 1, m \in M, \\ ST(s, n) &= ST(s, n-1) + \sum_{i \in I_s^p} \rho_{is}^p \\ &\quad \cdot \sum_{\substack{n' \in N \\ n-1-\Delta n \leq n' \leq n-1}} B^{(m)}(i, n', n-1) + \sum_{i \in I_s^c} \rho_{is}^c \\ &\quad \cdot \sum_{\substack{n'' \in N \\ n \leq n'' \leq n + \Delta n}} B^{(m)}(i, n, n''), \quad (15) \\ \forall s \in S^I, n \in N, n > 1, m \in M, \end{aligned}$$

$$\begin{aligned} ST(s, n) &= ST(s, n-1) + \sum_{i \in I_s^p} \rho_{is}^p \\ &\quad \cdot \sum_{\substack{n' \in N \\ n-1-\Delta n \leq n' \leq n-1}} B^{(m)}(i, n', n-1), \\ \forall s \in S^P, n \in N, n > 1, m \in M. \end{aligned}$$

3.1.4. Duration Constraints. If $\Delta n = 0$, the finish time of task i at event point n is equal to the start time plus duration time of the task, which is calculated by a constant term $\alpha_i w(i, n, n)$ and a variable term $\beta_i B^{(m)}(i, n, n)$:

$$T^f(i, n) = T^s(i, n) + \alpha_i w(i, n, n) + \beta_i B^{(m)}(i, n, n), \quad (16)$$

$$\forall i \in I, n \in N, \Delta n = 0, m \in M.$$

If $\Delta n > 0$, the duration constraints are given jointly by (17) and (18). If task i is active from event point n to event point n' , the finish time of task i at event point n' is equal to the sum of start time of task i at event point n and duration time of the task. Otherwise, (18) is relaxed, and finish time is greater than or equal to the start time:

$$T_f(i, n') \geq T^s(i, n) + \alpha_i w(i, n, n') + \beta_i B^{(m)}(i, n, n'), \quad (17)$$

$$\forall i \in I, n, n' \in N, n \leq n' \leq n + \Delta n, \Delta n > 0, m \in M,$$

$$T_f(i, n') \leq T^s(i, n) + \alpha_i w(i, n, n') + \beta_i B^{(m)}(i, n, n') + Q(1 - w(i, n, n')), \quad (18)$$

$$\forall i \in I, n, n' \in N, n \leq n' \leq n + \Delta n, \Delta n > 0, m \in M.$$

3.1.5. Sequencing Constraints

(1) *Same Task in the Same Unit.* The constraint for same task in the same unit is given by (19) for $\Delta n = 0$ and by (19) and (20) for $\Delta n > 0$. These two constraints enforce the finish time at current event to be equal to the start time at next event when the task is active and not ending at current event n :

$$T^s(i, n+1) \geq T^f(i, n), \quad \forall i \in I, n \in N, n < N, \quad (19)$$

$$T^s(i, n+1)$$

$$\leq T^f(i, n)$$

$$T^s(i, n+1) \geq T^f(i', n) - Q \left(1 - \sum_{\substack{n' \in N \\ n - \Delta n \leq n' \leq n}} w(i', n', n) \right),$$

$$\forall s, i, i', j, j', n \in N, n < N, i \in I_j, i' \in I_{j'}, i \neq i', j \neq j', i \in I_s^c, i' \in I_s^p.$$

For different tasks that produce or consume the same state, according to the production recipe, the start time of the consuming task at the next event is enforced to be later than the finish time of the producing task at the current event as given by (22). Otherwise, the constraint is relaxed and the two times are not related.

(2) *Different Tasks in the Same Unit.* Consider

$$T^s(i, n+1) \geq T^f(i', n), \quad (21)$$

$$\forall i, i' \in I_j, i \neq i', j \in J, n \in N, n < N.$$

Equation (21) is written for two different tasks performed in the same unit at event points n and $n+1$, respectively. If task i is active at event point $n+1$, the start time of task i must be greater than or equal to the finish time of that previous task i' .

(3) *Different Tasks in Different Units.* Consider

3.1.6. *No-Wait Condition.* The constraint is written for different tasks i and i' that produce or consume the same state consecutively with the “zero-wait” (ZW) condition because of storage restrictions on the intermediate material. Combined with (22), the two constraints enforce that task i in unit j at event point $n+1$ starts immediately after the end of task i' in unit j' at event point n if both tasks are activated:

$$T^s(i, n+1) \leq T^f(i', n) + Q \left(2 - \sum_{\substack{n' \in N \\ n - \Delta n \leq n' \leq n}} w(i', n', n) - \sum_{\substack{n' \in N \\ n+1 \leq n' \leq n+1+\Delta n}} w(i, n+1, n') \right), \quad (23)$$

$$\forall s \in S^{\text{dfis}}, S^{\text{zw}}, j, j' \in J, n < N, i \in I_j, i' \in I_{j'}, i \neq i', j \neq j', i \in I_s^c, i' \in I_s^p.$$

3.1.7. *Storage Constraints.* In this work, infinite storage and dedicated finite intermediate storage policies are considered. Equation (24) expresses the limit on storage capacity. For handling dedicated finite intermediate storage case, (25) is

used to avoid real time storage violations without the need for considering storage as a separate task. This constraint needs not to be written at the last event since the produced material is allowed to wait in the same processing unit at the last event:

$$ST(s, n) \leq ST_s^{\text{max}}, \quad \forall s \in S^{\text{dfis}}, n \in N, \quad (24)$$

$$T^f(i', n) \geq T^s(i, n) - Q \left(1 - \sum_{\substack{n' \in N \\ n - \Delta n \leq n' \leq n}} w(i', n', n) \right), \quad (25)$$

$$\forall s \in S^{\text{dfis}}, j, j' \in J, n < N, i \in I_j, i' \in I_{j'}, i \neq i', j \neq j', i \in I_s^c, i' \in I_s^p.$$

3.1.8. *Tightening Constraint.* Consider

$$\sum_{i \in I_j} \sum_{n \in N} \sum_{n' \in N} (\alpha_i w(i, n, n') + \beta_i B^{(m)}(i, n, n')) \leq H, \quad (26)$$

$$\forall j \in J, n \leq n' \leq n + \Delta n.$$

The sum of the durations of all tasks suitable in each unit should be less than the scheduling horizon as given by (26).

3.1.9. *Bounds on Variables.* Consider

$$\begin{aligned} w(i, n, n') &= 0; \\ B^{(m)}(i, n, n') &= 0, \end{aligned} \quad (27)$$

$$\forall n, n' \in N, n' < n, m \in M,$$

$$\begin{aligned} T^s(i, n) &\leq H; \\ T^f(i, n) &\leq H, \end{aligned} \quad (28)$$

$$\forall i \in I, n \in N.$$

The nonpermissible cases of the three-index binary and continuous variables are eliminated in (27), so that the corresponding variables are fixed to zero. General bounds are added to different continuous variables in (28). The start and finish times of all processing tasks must be within the time horizon.

3.1.10. *Objective Function: Maximization of Profit.* Consider

$$\begin{aligned} \text{Max profit}^{(m)} &= \sum_{s \in S^p} P_s \sum_{n \in N} \left(ST(s, n) \right. \\ &\quad \left. + \sum_{i \in I_s^p} \rho_{is}^p \sum_{\substack{n' \in N \\ n - \Delta n \leq n' \leq n}} B^{(m)}(i, n', n) \right). \end{aligned} \quad (29)$$

The objective function is the maximization of profit, and the total amount of the final products produced by the end of last event is calculated by (29).

3.2. *Lower Layer Model.* The input of the lower layer model is the scheduled processing amount of materials and the start and finish times of tasks from upper layer.

3.2.1. *Resource Constraint.* Consider

$$\begin{aligned} \sum_{i \in I_u} (\gamma_{i,u,sl} \cdot w(i, sl) + \delta_{iu} \cdot bs^{(m)}(i, sl)) \\ \leq U_u^{\max} \cdot \text{length}_{sl}, \end{aligned} \quad (30)$$

$$\forall u \in U, i \in I_u, sl \in \text{Slot}, m \in M.$$

The utility utilization level at slot sl is the sum of amounts used by tasks that consume the utility at slot sl , which is

composed of a constant term $\gamma_{i,u,sl} \cdot w(i, sl)$ and a variable term $\delta_{iu} \cdot bs^{(m)}(i, sl)$. The utility utilization level is limited by the total supply of utility and duration time of slot sl . In this work, parameter length_{sl} is introduced to present the duration time of slot, which makes the supply of utility increase with time rather than constant.

3.2.2. *Batch Constraints.* Consider

$$\sum_{sl \in \text{Slot}} bs^{(m)}(i, sl) \leq B_{i,sl',sl''}^{(m)}, \quad (31)$$

$$sl' \leq sl \leq sl'', (i, sl', sl'') \in \text{Task}, m \in M,$$

$$\text{diffs}^{(m)}(i, sl', sl'') = B_{i,sl',sl''}^{(m)} - \sum_{sl \in \text{Slot}} bs^{(m)}(i, sl), \quad (32)$$

$$sl' \leq sl \leq sl'', (i, sl', sl'') \in \text{Task}, m \in M.$$

Batch constraints express a limit that the assignments $B_{i,sl',sl''}^{(m)}$ from upper layer scheduling solution should be greater than or equal to the processing amount of task i within corresponding slots. Because the event points in upper lower are divided into several slots, there is a mapping between event points and slots. As shown in Figure 5, for unit 3, the event point N_1 starts from $T^s(i_1, N_1)$ and ends at $T^f(i_1, N_1)$, and the event point N_2 starts from $T^s(i_1, N_2)$ and ends at $T^f(i_1, N_2)$, where $T^f(i_1, N_1) = T^s(i_1, N_2)$. Event points N_1 and N_2 are divided into four slots, so the mapping between event and slot is $\text{event}(n_1, n_2) = \text{slot}(sl_1, sl_4)$, also $B(i, n_1, n_2) = b(i, sl_1) + b(i, sl_2) + b(i, sl_3) + b(i, sl_4)$. Here, in the m th iteration, variable $B^{(m)}(i, n, n')$ from upper layer is equal to parameter $B_{i,sl',sl''}^{(m)}$ in (31) in lower layer.

3.2.3. *Capacity Constraint.* The capacity constraint of batch size should also be involved in the lower layer model. The formulation is similar with that in the upper layer model:

$$\begin{aligned} B_i^{\min} \frac{\text{length}_{sl}^{(m)}}{\text{length_event}_{i,sl',sl''}^{(m)}} &\leq bs^{(m)}(i, sl) \\ &\leq B_i^{\max} \frac{\text{length}_{sl}^{(m)}}{\text{length_event}_{i,sl',sl''}^{(m)}}, \end{aligned} \quad (33)$$

$$\forall i \in I, sl \in \text{slot}, sl' \leq sl \leq sl'', (i, sl', sl'') \in \text{Task}.$$

3.2.4. *Objective Function*

$$\begin{aligned} \text{Min}^{(m)} &= \text{result}^{(m)} + \xi \cdot \sum [\text{smooth}^{(m)}(i, sl', sl'')], \\ &\quad (i, sl', sl'') \in \text{Task}, \end{aligned}$$

$$\begin{aligned} \text{result}^{(m)} &= \sum \left[B_{i,sl',sl''}^{(m)} - \sum_{sl' \leq sl \leq sl''} bs^{(m)}(i, sl) \right], \\ &\quad (i, sl', sl'') \in \text{Task}, \end{aligned}$$

$$\begin{aligned}
& \text{smooth}^{(m)}(i, sl', sl'') \\
&= \sum \left[\frac{bs^{(m)}(i, sl+1)}{\text{length}_{sl+1}^{(m)}} - \frac{bs^{(m)}(i, sl)}{\text{length}_{sl}^{(m)}} \right]^2, \\
& \quad (i, sl', sl'') \in \text{Task}, \quad sl' \leq sl < sl''.
\end{aligned} \tag{34}$$

The objective of lower layer is to complete the scheduled processing amount of materials within the scheduled time which is from upper layer calculated solutions under utility constraints. Thus, the objective function is to minimize the difference between the scheduled processing amount of materials from upper layer and the processing amount in lower layer. The last term in the objective function is a penalty on fluctuation of production rate. Coefficient ξ is a penalty term and set to 0.0001.

3.3. Iteration between Upper and Lower Layers. The optimal solution of the proposed model is found through iterations between upper layer and lower layer. Firstly, production scheduling problem is solved by upper layer model without utility consideration. The solutions of upper layer are allocated to the lower layer as assignments, which include processing amount of materials in each event point $B(i, n, n')$ and the start and finish times of tasks at each event point $T^s(i, n)$ and $T^f(i, n)$. The objective of the lower layer is to complete the scheduled processing amount of materials within the scheduled time which is from upper layer calculated solutions under utility constraints. If the first term of objective function result^(m) in lower layer is equal to zero, the scheduling solution from upper layer is feasible under utility limitations. Otherwise, the assignments cannot be completed with utility constraints. The parameter $\text{Diff}_{i,n,n'}^{(m)}$ will be calculated and fed back to the upper layer. The flowchart diagram is shown as in Figure 7.

4. Computational Case Studies

In this section, we choose two examples from the existing literatures involving short-term scheduling of batch plants with resource consideration. Both two examples are given by STN representation, solved using Lingo 11.0 with a 2.20 GHz computer. The effectiveness and the advantages of the proposed hierarchical model are discussed in Sections 4.1 and 4.2.

4.1. Example 1. The first example is introduced firstly by Maravelias and Grossmann and has been studied in several literatures [12, 18, 19]. As shown in Figure 8, this example involves four tasks in three units (reactors) processing seven states. The corresponding data are given in Tables 1, 2, and 3. There are two types of reactors available for the process (types I and II) with different number of corresponding units available: two reactors (RI1 and RI2) of type I and one reactor (RII) of type II. Reactions T1 and T2 require type I reactor, while reactions T3 and T4 require type II reactor. A task

FIGURE 7: Flowchart diagram for iterations.

FIGURE 8: State-task network representation for example 1.

which can be performed in different units is considered as multiple, separate tasks, thus leading to six separate tasks ($i = 1, \dots, 6$), each suitable in one unit. Furthermore, reactions T1 and T3 require heat, provided by steam (HS) produced within limited amounts in the plant, while reactions T2 and T4 are exothermic and require cooling water (CW), also available in limited amounts. Due to safety reasons and temperature restrictions, the heat integration of the process is not possible.

This example includes variable batch sizes, processing time, and utility requirements. The minimum batch size for all tasks is the half of maximum capacities of units. Processing times for all tasks are functions of the batch size. Utility requirements include a fixed term as well as a variable term proportional to the batch size. The utility available is assumed to be 30 kg/min. For the raw materials (F1 and F2) and final products (P1 and P2), unlimited storage is available

FIGURE 9: Gantt chart and utility utilization levels for proposed model in example 1.

FIGURE 10: Gantt chart and utility utilization levels for V&S model in example 1.

(say 1000 kg), while for the intermediates (I1, I2, and I3), finite storage is available.

In example 1, scheduling horizon is 8 hours and the maximum number of event points Δn is set to 1. Solution of this example is compared with example 10b which is studied by V&S [18]. The objective function of example 1 is to maximize profits. The optimal solution of example 1 without utility consideration is 6289.583\$, and the optimal solution of V&S model is 5227.778\$ under utility consideration. The optimal solution of the proposed model is 5525.000\$ under

utility consideration with six events and with higher events as well. The optimal production schedule and utility utilization levels of the proposed model and V&S model are, respectively, provided in Figures 9 and 10.

The optimal production schedule of the proposed model is illustrated by the Gantt chart of units in Figure 9. Levels of utilization of steam and cooling water at the optimal solution are given in the plots in Figure 9. Lines in Gantt charts show each event point at which a task is active. Values below the lines are numbers of tasks, and values above the lines

TABLE 1: Data of production capability and coefficients of processing time for example 1.

	$B^{\min}$	$B^{\max}$	T1		T2		T3		T4	
			α	β	α	β	α	β	α	β
R1	40	80	0.5	0.025	0.75	0.0375	—	—	—	—
R2	25	50	0.5	0.04	0.75	0.06	—	—	—	—
R3	40	80	—	—	—	—	0.25	0.0125	0.5	0.025

$B^{\min}, B^{\max}$ in kg, α in h, and β in h/kg.

TABLE 2: Data of coefficients of utility consumption for example 1.

	T1		T2		T3		T4	
	γ_{iHS}	δ_{iHS}	γ_{iCW}	δ_{iCW}	γ_{iHS}	δ_{iHS}	γ_{iCW}	δ_{iCW}
R1	6	0.25	4	0.3	—	—	—	—
R2	4	0.25	3	0.3	—	—	—	—
R3	—	—	—	—	8	0.2	4	0.5

γ_{iHS} in kg and δ_{iHS} in kg per kg of batch.

TABLE 3: State related data for example 1.

	F1	F2	I1	I2	I3	P1	P2
$ST_s^{\max}$ (kg)	1000	1000	200	100	500	1000	1000
ST_s^0 (kg)	400	400	0	0	0	0	0
Price _s (\$/kg)	0	0	0	0	0	30	40

are processing amount of corresponding tasks. As shown in Figure 9, 77.5 kg of P1 and 80 kg of P2 are produced. The objective function of maximizing profits is 5525\$. Comparing with the V&S model, 55 kg of P1 and 89.4 kg of P2 are produced, as shown in Figure 10. The objective function of maximizing profits in V&S model is 5227.778\$. The reason for the less profit of V&S model is caused by multiple counting problems mentioned in Problem Statement section. V&S model allows tasks to extend over more than one event point. The same task extending over two adjacent event points appears as two separate tasks, each with the same batch size. Also, resource utilization levels are determined at each event point and thus do not reflect the actual utility consumption levels. As shown in Figure 10, task 4 extends over two event points; that is, task 4 starts at event point N_1 and ends at event point N_2 . The consumption of CW for task 4 from event point N_1 to event point N_2 is counted in event point N_1 and counted repeatedly in event point N_2 , which leads the utilization level of CW during the first interval to reach the maximum 30 kg/min. Moreover, the consumption of CW for task 4 at the last half of second interval should be zero. The shaded area shown in the utility utilization levels for CW in Figure 10 is an unreal statistics. The unreal statistics will lead to an unreal shortage of utility. Thus, the processing amount of optimal schedule calculated by S&V model is limited by the unreal shortage of utility.

In the proposed model, event points of different units are mapped into a uniform time presentation, which avoids the overestimation of utility utilization. As shown in Figure 9, the consumption of CW in slot 3 is zero.

FIGURE 11: Trajectories for searching the optimal solution in example 1.

In the V&S model, utility requirements include a fixed term as well as a variable term proportional to the batch size, as given by

$$\sum_{i \in I_u} \left(\gamma_{iu} \sum_{n' \in N} \sum_{n'' \in N} w(i, n', n'') \right) + \sum_{i \in I_u} \left(\delta_{iu} \sum_{n' \in N} \sum_{n'' \in N} b(i, n', n'') \right) \leq U_u^{\max}, \quad (35)$$

$$\forall u \in U, n \in N, n - \Delta n \leq n' \leq n, n \leq n'' \leq n' + \Delta n.$$

The utility requirements in the proposed model also consist of a fixed term as well as a variable term. According to the mapping relationship between event(n, n') and slot(sl, sl'), there is a relationship between γ_{iu}, δ_{iu} in (35) and $\gamma_{i,u,sl}, \delta_{iu}$ in (30):

$$\gamma_{i,u,sl} = \frac{\gamma_{iu}}{sl'' - sl' + 1}, \quad u \in U, i \in I_u, sl' \leq sl \leq sl'', (i, sl', sl'') \in \text{Task}, \quad (36)$$

$$\delta_{iu} = \delta_{iu},$$

$$u \in U, i \in I_u, sl' \leq sl \leq sl'', (i, sl', sl'') \in \text{Task}.$$

The trajectories for searching the optimal solution of upper layer and lower layer are shown in Figure 11 and the data are given in Table 4. Firstly, upper layer searches the optimal solution without involving resource constraints and then puts solutions into lower layer model. The iterations between upper and lower layers terminate when no further improvement for the objective function can be achieved. Through eight iterations, the optimal solution 5525.000\$ is found.

TABLE 4: Optimization solution of upper and lower layers in example 1.

Iteration number	Objective of upper layer \$	Objective	Solution of lower layer Result	Smooth
1	6289.583	23.92516	23.783787	1413.785
2	5960.000	4.5	4.5	0
3	5960.000	12	12	0
4	5960.000	4.5	4.5	0
5	5915.000	10.5	10.5	0
6	5600.000	3.166667	3.166667	0
7	5600.000	7.5	7.5	0
8	5525.000	0.3900007E - 07	0.3900007E - 07	0

FIGURE 12: State-task network representation for example 2.

4.2. Example 2. Example 2 is a standard case for short-term scheduling of multipurpose batch plants and has been studied frequently by many authors [17, 18]. Two different products are produced through five processing stages: heating, reactions 1, 2, and 3, and separation, as shown in the STN representation of the plant flow sheet in Figure 12. Since each of the reaction tasks can take place in two different reactors, each reaction is represented by two separate tasks. In order to demonstrate the effectiveness of proposed model, we suppose that all reactions 1, 2, and 3 require a certain utility. The utility available is assumed to be 200 kg/min. The relevant data are shown in Table 5. The initial inventory level for all intermediates is assumed to be zero and unlimited storage capacity is assumed for all states. The prices of products 1 and 2 are \$10/mu.

In this example, scheduling horizon is 10 hours and the maximum number of event points Δn is set to 1. Solution of this example is compared with example 2b studied by V&S [18]. The optimal solution of example 2 without utility consideration is 1962.695\$, and the optimal solution of V&S model is 1528.208\$ under utility constraints. The optimal solution of the proposed model in this work is 1928.526\$ under utility constraints with six events and with higher events as well. The optimal production schedule and utility

TABLE 5: Data for example 2.

Task (i)	Unit (j)	α_i	β_i	γ_i	δ_i	$B_i^{\min}$	$B_i^{\max}$
Heating ($i = 1$)	Heater	0.667	0.00667	—	—	0	100
Reaction 1 ($i = 2$)	Reactor 1	1.334	0.02664	4	3	0	50
($i = 3$)	Reactor 2	1.334	0.01665	6	5	0	80
Reaction 2 ($i = 4$)	Reactor 1	1.334	0.02664	6	3	0	50
($i = 5$)	Reactor 2	1.334	0.01665	4	2	0	80
Reaction 3 ($i = 6$)	Reactor 1	0.667	0.01332	2	1	0	50
($i = 7$)	Reactor 2	0.667	0.008325	6	4	0	80
Separation ($i = 8$)	Separator	1.3342	0.00666	—	—	0	200

$B_i^{\min}$, $B_i^{\max}$ in mu, α in h, β in h/mu, γ_{HS} in kg, and δ_{HS} in kg per mu of batch.

utilization levels of the proposed model and V&S model are provided in Figures 13 and 14.

As shown in Figure 13, 78.10 mu of P1 and 114.75 mu of P2 are produced. The objective function of maximizing profits is 1928.526\$. Comparing with the V&S model, 61.72 mu of P1 and 91.08 mu of P2 are produced, as shown in Figure 14. The objective function of maximizing profits in the V&S model is 1528.208\$. The less profit of V&S model in example 2 is caused by overstrict constraint problems and multiple

FIGURE 13: Gantt chart and utility utilization levels for proposed model in example 2.

FIGURE 14: Gantt chart and utility utilization levels for V&S model in example 2.

counting problems mentioned in Problem Statement section. The two discrepancies of V&S model both appear in this case. As shown in the Gantt chart of Figure 14, one discrepancy is derived from forcing the start time of task 6 and task 7 at event point N_5 to be equal, which leads to a large batch size processed in event point N_5 and results in a resource shortage. Compared with proposed model in this work, the start times of task 6 and task 7 at event point N_5 are not forced to be equal, as shown in the Gantt chart of Figure 13. The other discrepancy is double counting of utility utilization. As shown in the Gantt chart of Figure 14, task 5 continues over two event points. It starts at event point N_3 and ends at event points N_4 . The utility utilization level of task 5 is counted at event point N_3 and counted repeatedly at event point N_4 . This miscounting of utility utilization results in the fact that other tasks active at event N_3 and N_4 cannot get enough resource,

which may also result in the reduction of processing amount and further lead to less profit.

The trajectories for searching the optimal solution of upper layer and lower layer are shown in Figure 15 and the data are given in Table 6. Firstly, upper layer searches the optimal solution without considering resource constraints and then puts the solutions into lower layer model. Through six iterations, the optimal solution 1928.526\$ is found.

5. Conclusions

In this work, some limitations of continuous-time unit-specific event-based model with utility constraints are stated, including the following: (a) overstrict constraint problem about utility resource, (b) multiple counting problems, and (c) missing of duration time of event points. To solve

TABLE 6: Optimization solution of upper and lower layers in example 2.

Iteration number	Objective of upper layer \$	Solution of lower layer		
		Objective	Result	Smooth
1	1962.695	5.757539	5.757539	0
2	1938.856	1.705364	1.705364	0
3	1931.919	34.15167	34.15154	1.3498342
4	1931.856	0.5962500	0.5962500	0
5	1929.287	0.1764063	0.1764063	0
6	1928.526	0.4399189E - 07	0.4399189E - 07	0

FIGURE 15: Trajectories for searching the optimal solution in example 2.

Sets

- I : Tasks
 I_j : Tasks which can be performed in unit j
 I_s^P : Tasks which produce state s
 I_s^C : Tasks which consume state s
 I_u : Tasks which consume utility u
 J : Units
 J_i : Units which are suitable for performing task i
 N : Total event points postulated in the time horizon
 S : All states
 S^R : States that are raw materials
 S^P : States that are final products
 S^I : States that are intermediates
 U : Utilities
 M : Iteration numbers
 $S^{\text{dis}}, S^{\text{zw}}$: Intermediate states with dedicated finite intermediate storage and zero wait condition, respectively.

these problems, a framework with hierarchical structure is proposed. In upper layer, a unit-specific event-based model is used, and a slot-based model is introduced in lower layer to jointly address the short-term production scheduling problem of batch plants under utility consideration. The effectiveness and advantages of proposed model are illustrated by two benchmark examples in the existing literatures.

Nomenclature

Upper Layer

Subscripts

- i, i' : Task
 j, j' : Unit
 n, n', n'' : Event
 s : State
 u : Utility
 sl : Time slot
 m : Iteration number.

Parameters

- $B_i^{\min}$: Minimum batch size of task i
 $B_i^{\max}$: Maximum batch size of task i
 ST_s^0 : Initial amount available for state s
 $ST_s^{\max}$: Maximum storage capacity for state s
 $\text{Diff}_{i,n,n'}^{(m)}$: Feedback of task i that starts at event n and ends at event n' from lower layer which is added in capability constraints in upper layer, when the m th iteration
 ρ_{is}^c : Proportion of state s consumed by task i , $\rho_{is}^c \leq 0$
 ρ_{is}^p : Proportion of state s produced by task i , $\rho_{is}^p \geq 0$
 α_i : Fixed processing time of task i
 β_i : Linear coefficient of the variable term of the processing time of task i
 P_s : Price of state s
 Δn : Limit on the maximum number of events over which a task is allowed to continue

Q: Large positive number in big-Q constraints

H: Short-term scheduling horizon.

Binary Variables

$w(i, n, n')$: Binary variable for task i that starts at event n and ends at event n' .

Positive Variables

$B^{(m)}(i, n, n')$: Amount of material processed by task i starting at event n and ending at event n' in the m th iteration, which has a mapping relationship with parameter $B_{i,sl',sl''}^{(m)}$ in lower layer

$ST(s, n)$: Excess amount of state s that needs to be stored at event n

$T^s(i, n)$: Start time of a task i at event n

$T^f(i, n)$: End time of a task i at event n .

Lower Layer

Subscripts

i : Task

u : Utility

sl : Time slot

m : Iteration number.

Sets

I : Tasks

I_u : Tasks which consume utility u

U : Utilities

Slot: Time slots

Task: A subset of tasks and slots (i, sl', sl'') which matches set (i, n, n') where task i is active from event n to event n' in upper layer

M : Iteration numbers.

Parameters

$\gamma_{i,u,sl}$: Coefficient of constant term for utility u consumption by task i in slot sl

δ_{iu} : Coefficient of variable term for utility u consumption by task i

$U_u^{\max}$: Maximum availability of utility u

$\text{length}_{sl}^{(m)}$: Duration time of slot sl in the m th iteration

$\text{length_event}_{i,sl',sl''}^{(m)}$: Duration time of event at which task i is active in upper layer and corresponding to the duration time from slot sl' to slot sl'' in lower layer at the m th iteration

$B_{i,sl',sl''}^{(m)}$: Assignment from upper layer for task i active from slot sl' to slot sl'' in the m th iteration, which has a mapping relationship with $B^{(m)}(i, n, n')$

ξ : Penalty term for avoiding fluctuation of production rate.

Binary Variables

$w(i, sl)$: Binary variable for task i that is active at slot sl .

Positive Variables

$\text{diffs}^{(m)}(i, sl', sl'')$: Backlog of task i from slot sl' to slot sl'' in the m th iteration, which has a mapping relationship with $\text{diffs}^{(m)}(i, n, n')$

$bs^{(m)}(i, sl)$: Amount of material processed by task i at slot sl in the m th iteration

$\text{smooth}^{(m)}(i, sl', sl'')$: Statistics for smoothness of production rate in the m th iteration

$\text{result}^{(m)}$: Statistics for completion of assignment from upper layer in the m th iteration.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This project is supported by the National High-Tech R&D Program of China (863 Program: 2014AA041803) and the National Natural Science Foundation of China (NSFC: 61134007).

References

- [1] A. B. Keha, K. Khowala, and J. W. Fowler, "Mixed integer programming formulations for single machine scheduling problems," *Computers & Industrial Engineering*, vol. 56, no. 1, pp. 357–367, 2009.
- [2] W. Cheng, P. Guo, Z. Zhang, M. Zeng, and J. Liang, "Variable neighborhood search for parallel machines scheduling problem with step deteriorating jobs," *Mathematical Problems in Engineering*, vol. 2012, Article ID 928312, 20 pages, 2012.
- [3] M. Joly and J. M. Pinto, "Mixed-integer programming techniques for the scheduling of fuel oil and asphalt production," *Chemical Engineering Research and Design*, vol. 81, no. 4, pp. 427–447, 2003.
- [4] J. Cheng, J. Fowler, K. Kempf, and S. Mason, "Multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting," *Computers & Operations Research*, vol. 53, pp. 275–287, 2015.
- [5] C. A. Méndez and J. Cerdá, "An MILP framework for short-term scheduling of single-stage batch plants with limited discrete resources," *Computer Aided Chemical Engineering*, vol. 10, pp. 721–726, 2002.
- [6] C. A. Méndez and J. Cerdá, "An MILP framework for batch reactive scheduling with limited discrete resources," *Computers & Chemical Engineering*, vol. 28, no. 6–7, pp. 1059–1068, 2004.

- [7] J. M. Pinto and I. E. Grossmann, "A logic-based approach to scheduling problems with resource constraints," *Computers & Chemical Engineering*, vol. 21, no. 8, pp. 801–818, 1997.
- [8] J. Zhang, W. Wang, and X. Xu, "A hybrid discrete particle swarm optimization for dual-resource constrained job shop scheduling with resource flexibility," *Journal of Intelligent Manufacturing*, 2015.
- [9] A. Lindholm and P. Giselsson, "Minimization of economical losses due to utility disturbances in the process industry," *Journal of Process Control*, vol. 23, no. 5, pp. 767–777, 2013.
- [10] A. Lindholm and N.-P. Nytzén, "Hierarchical scheduling and disturbance management in the process industry," *Computers & Chemical Engineering*, vol. 71, pp. 489–500, 2014.
- [11] Y. Wang, H. Y. Su, S. Lu, L. Xie, and L. Zhu, "Hierarchical approach of planning and scheduling with demand uncertainty and utility disturbance," in *Proceedings of the 5th International Symposium on Advanced Control of Industrial Processes (ADCONIP '14)*, Hiroshima, Japan, May 2014.
- [12] S. L. Janak, X. Lin, and C. A. Floudas, "Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: resource constraints and mixed storage policies," *Industrial & Engineering Chemistry Research*, vol. 43, no. 10, pp. 2516–2533, 2004.
- [13] B. Behdani, M. R. Pishvaie, and D. Rashtchian, "Optimal scheduling of mixed batch and continuous processes incorporating utility aspects," *Chemical Engineering and Processing: Process Intensification*, vol. 46, no. 4, pp. 271–281, 2007.
- [14] N. Susarla, J. Li, and I. A. Karimi, "A novel multi-grid formulation for scheduling semi-continuous plants," *Computer Aided Chemical Engineering*, vol. 31, pp. 1075–1079, 2012.
- [15] M. A. Shaik and R. Vooradi, "Unification of STN and RTN based models for short-term scheduling of batch plants with shared resources," *Chemical Engineering Science*, vol. 98, pp. 104–124, 2013.
- [16] C. A. Méndez and J. Cerdá, "Short-term scheduling of multistage batch processes subject to limited finite resources," *Computer Aided Chemical Engineering*, vol. 15, pp. 984–989, 2003.
- [17] M. A. Shaik and C. A. Floudas, "Novel unified modeling approach for short-term scheduling," *Industrial & Engineering Chemistry Research*, vol. 48, no. 6, pp. 2947–2964, 2009.
- [18] R. Vooradi and M. A. Shaik, "Improved three-index unit-specific event-based model for short-term scheduling of batch plants," *Computers & Chemical Engineering*, vol. 43, pp. 148–172, 2012.
- [19] C. T. Maravelias and I. E. Grossmann, "New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants," *Industrial & Engineering Chemistry Research*, vol. 42, no. 13, pp. 3056–3074, 2003.
- [20] A. Sundaramoorthy and I. A. Karimi, "A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants," *Chemical Engineering Science*, vol. 60, no. 10, pp. 2679–2702, 2005.