AIChE

PROCESS SYSTEMS ENGINEERING

Moving Horizon Approach of Integrating Scheduling and Control for Sequential Batch Processes

Yunfei Chu and Fengqi You
Dept. of Chemical and Biological Engineering, Northwestern University, Evanston, IL 60208

DOI 10.1002/aic.14359
Published online January 23, 2014 in Wiley Online Library (wileyonlinelibrary.com)

Online integration of scheduling and control is crucial to cope with process uncertainties. We propose a new online integrated method for sequential batch processes, where the integrated problem is solved to determine controller references rather than process inputs. Under a two-level feedback loop structure, the integrated problem is solved in a frequency lower than that of the control loops. To achieve the goal of computational efficiency and rescheduling stability, a moving horizon approach is developed. A reduced integrated problem in a resolving horizon is formulated, which can be solved efficiently online. Solving the reduced problem only changes a small part of the initial solution, guaranteeing rescheduling stability. The integrated method is demonstrated in a simulated case study. Under uncertainties of the control system disruption and the processing unit breakdown, the integrated method prevents a large loss in the production profit compared with the simple shifted rescheduling solution. © 2014 American Institute of Chemical Engineers *AIChE J*, 60: 1654–1671, 2014

Keywords: scheduling, batch processes, mixed-integer dynamic optimization, model predictive control, advanced manufacturing

Introduction

Enterprise-wide optimization has become a major goal in the process industries, which can account for complex tradeoffs and interactions across the various functions, subsystems, and levels of decision making.[1–4] As two major decision-making layers in the production hierarchy, integration of scheduling and control has attracted significant research efforts in recent years.[5–20] Compared with the traditional method where the scheduling problem and the control problem are solved sequentially, the integrated method can optimize the overall performance of the production process by making a better coordination between the subsystems.

However, most integrated methods focus on simultaneously optimizing the two problems offline. There are relatively few studies[8,12] which investigate how the integrated approach can be implemented online. There are three critical issues for an online integrated method, which are often neglected in existing methods.

The first issue is how the integrated method cooperates with unit controllers. In most existing methods, the controllers are completely incorporated into the integrated problem, which is solved to determine the process inputs directly. As shown in Figure 1a, the integration structure results in an open-loop control system. The integrated method solves a mixed-integer dynamic optimization (MIDO)[21,22] problem to optimize scheduling decisions along with the operational recipe, for example, processing times and processing costs. The integration structure completely neglects process uncertainties. However, unforeseen disruptions are inevitable in a process, which can cause the actual production to deviate from what is planned.[23–28] To cope with uncertainties, the integrated problem should be resolved online. Based on the open-loop structure in Figure 1a, a closed-loop integration, which is shown in Figure 1b, can be built by resolving the integrated problem at each controller sampling point according to measured real-time data.[12] Considering that the integrated problem is a plant-wide optimization including dynamic models for all possible combinations of operational tasks and processing units, the complicated MIDO problem is difficult to solve in a frequency as high as the controller sampling rate. To alleviate the computational burden, another integration framework has been developed,[8] which is shown in Figure 1c. The framework requires parameterized controllers, for example, PI controllers. The integrated problem is solved to update the controller parameters rather than the process inputs. Because the controllers work in real time, the processing units are always controlled in feedback loops. However, the requirement of parameterized controllers prevents implementing the integrated method with some advanced control techniques, like model predictive control (MPC).

Besides the integration structure, another issue which is often neglected is the computational efficiency. Most previous studies concentrate on the formulation of the integrated problem. The formulated MIDO problem is often discretized into a mixed-integer nonlinear programming (MINLP) problem which is then solved by a general-purpose MINLP solver.[9,12–14] This solution approach, called the simultaneous method, is straightforward. However, the computational time

Correspondence concerning this article should be addressed to F. You at you@northwestern.edu.

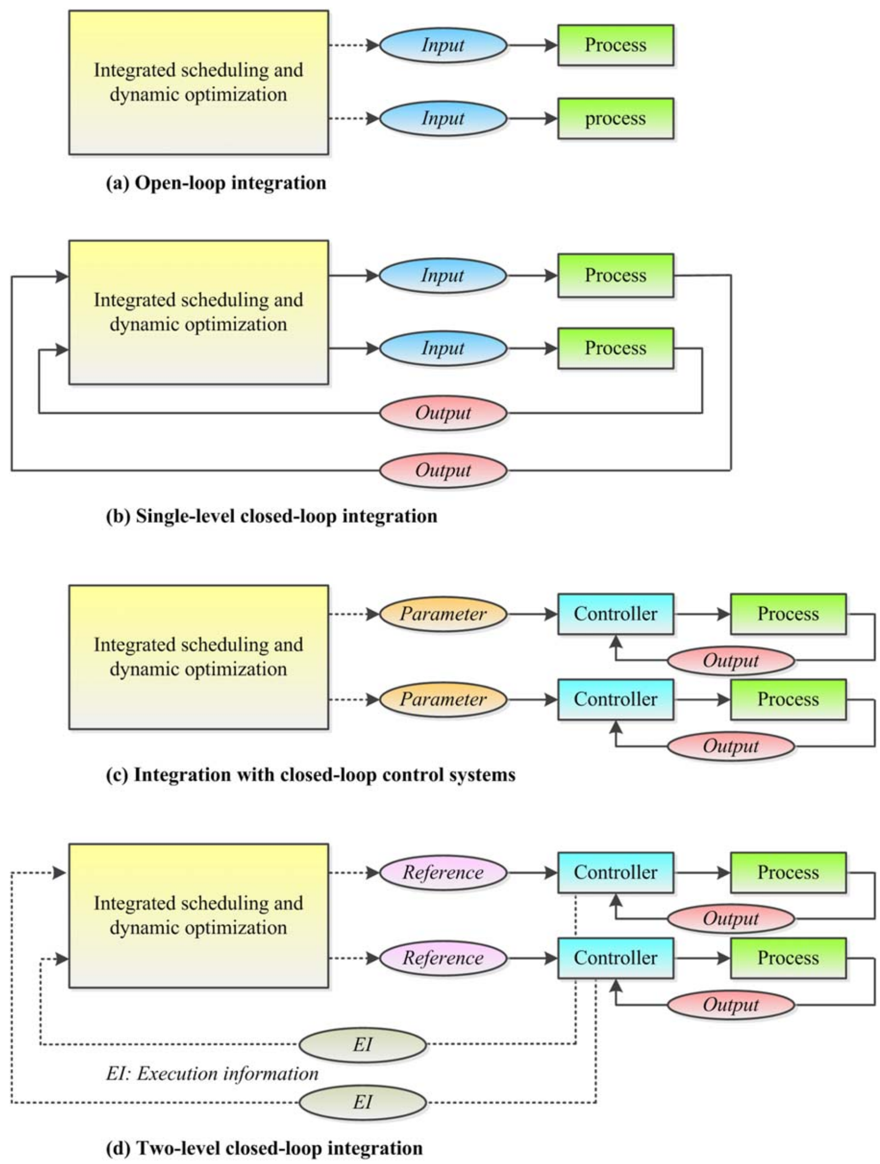


Figure 1. Structures for integrated scheduling and control problem.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

is generally too long for online implementation. Except the continuous process,[12] there is lack of the online simultaneous method for the batch process.

The third important, while often neglected, issue for an online integrated method is the rescheduling stability. It is a measure of the discrepancy between the initial schedule and the rescheduling solution.[29] Different from the automatic control loops, execution of a schedule usually requires human resources. Frequent and dramatic change in the rescheduling solution can cause the problem of "floor nervousness,"[30] substantially lowering the production efficiency. In addition, scheduling decisions are often linked to other decision levels like the supply chain planning. Large variations in the rescheduling solution can cause negative effects on the coordination with other decision levels. A good rescheduling approach should improve the rescheduling performance while prevent dramatic changes in the rescheduling solution.[31] The existing integrated methods seldom take the rescheduling stability into account, which tend to change the entire schedule even if only a minor disruption occurs.

To address the three important online implementation issues, we propose a new online integrated method. To cooperate with the advanced controllers, a novel integration framework is developed in Figure 1d. The integrated problem is solved to determine the reference trajectories for controllers. The controllers then track the references by manipulating the process inputs in the real-time feedback loops. Unlike the framework in Figure 1b, the controllers do not return the output trajectory to the higher level. Instead, execution information of completed tasks is returned. The execution information summarizes the results of control systems required by the scheduling model. Specifically, in this work, it includes the task processing times and processing costs that are calculated from the dynamic trajectories of the control systems under uncertainties. The deviation between the real measured values and the predetermined values reflects the uncertainties. At the higher level, the integrated problem is solved online to determine new reference trajectories along with scheduling decisions. The integrated framework contains two levels of feedback loops. We should note that the top-level rescheduling feedback loop works in a different time scale from the bottom-level control feedback loops. The rescheduling feedback loop is event-driven, for example, by task completion or unit breakdown. Thus, the integrated problem does not need to be solved in the same frequency as high as the controller loops. The different time scales facilitate online implementation of the integrated method.

Besides the integration structure, we develop a moving horizon approach to solve the integrated problem online, which addresses computational efficiency and rescheduling stability at the same time. Under uncertainties, this policy determines a new solution of the integrated problem based on the initial one (the offline solution or the previous online solution). Only a reduced problem in a short horizon is solved while the scheduling decisions and controller references beyond the horizon follow those of the initial solution. Dramatic change in the rescheduling solution is avoided as only a segment of the initial schedule is modified. Another benefit from the moving horizon policy is that only a reduced problem is solved which is much simpler than the entire problem.

The novelties of this work in solving integrated scheduling and control problem online to cope with uncertainties are summarized as

- A new online integration framework with the two-level feedback loops working in different time scales.
- A moving horizon predictive-reactive solution approach that guarantees rescheduling stability while reduces the computational time for a new online solution.
- A formulation of the reduced problem that can be solved efficiently online.

It should be noted that we use the moving horizon approach to solve the integrated problem online, which is distinct from those for solving scheduling problems.[32–34] Besides the scheduling model, the integrated problem includes a number of dynamic models. The online integrated problem determines not only scheduling decisions but also controller references. Processing times and processing costs are variables in the integrated problem while they are commonly fixed parameters in a conventional scheduling problem.

The remainder of the article is organized as follows. The problem statement and the integration framework have been explained in Problem Statement Section and Online Integration Framework Section, respectively. The core component of the framework is to solve a reduced problem formulated in Formulation of Online Integrated Problem Section. A Case Study Section demonstrates the proposed method. The conclusion is given in final section.

Problem Statement

This work concentrates on the sequential batch process where material splitting and mixing is prohibited. In the sequential batch process, the batch integrity is preserved and the batch size is fixed. The batch size is predetermined by solving a planning problem, often called the batching problem. As illustrated in Figure 2a, a customer order is divided into suborders, each of which can be fulfilled by one batch. The entire production procedure from the raw materials to the final products for fulfilling a suborder is called a job, which consists of several operational stages shown in Figure 2b. An operational stage can be executed in processing units specified by the job routing diagram. A job operational stage is also called a task, which is indicated by the combination of the job index and the stage index. In a sequential batch scheduling problem, the job information is commonly given directly and the batching procedure is not taken into account.

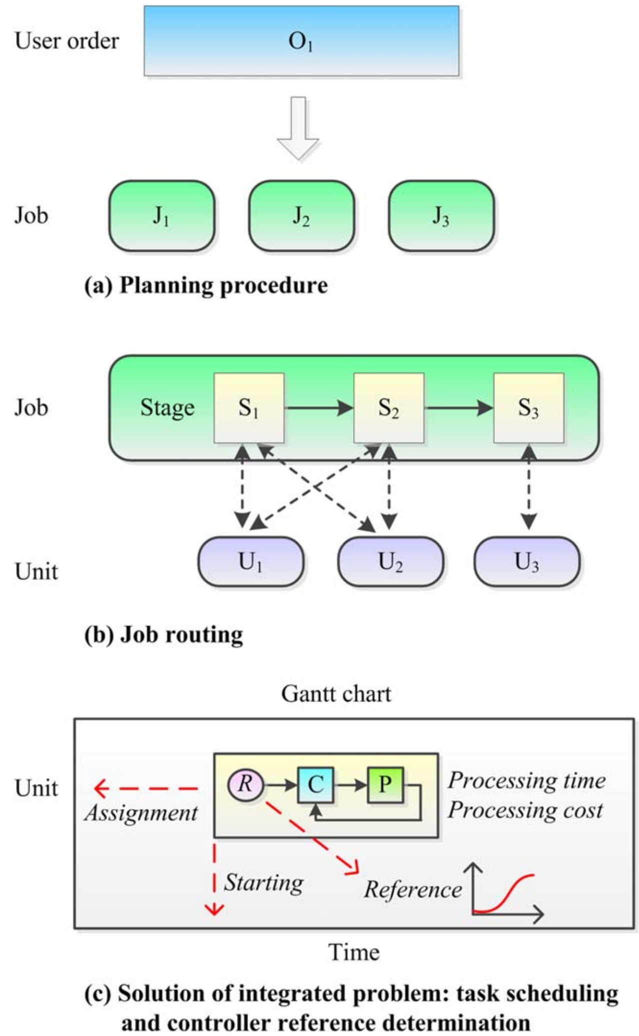


Figure 2. Statement of integrated scheduling and control problem.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

The integrated problem is solved to determine scheduling decisions and controller references. The scheduling decisions are often visualized by a Gantt chart illustrated in Figure 2c, which shows the assignment of a task to a processing unit, the starting time of the task, and the execution duration. Different form a conventional scheduling problem, the integrated problem can change the recipe data such as the processing time and the processing cost for a task. The recipe data are variables that can be manipulated by setting the reference trajectories for the control system associated with the task.

In the proposed integration framework, we assume a local controller can stabilize the unit dynamic system and have a good performance for tracking the reference trajectories determined by the integrated problem. The requirement on the stability and the tracking performance is actually a standard objective for designing a control system, imposing no restrictions on the integrated method. Of course, the references cannot be exactly tracked by a real controller. The discrepancy between the reference trajectories and the real ones is regarded as a type of uncertainties, which is fed back to the integrated problem along with other uncertainties. Then, the integrated problem is resolved online taking the uncertainties into account including the trajectory discrepancy.

Specifically, the online integrated scheduling and control problem is stated as

---

Implementation
- To solve the integrated scheduling and control problem online under uncertainties

Process structure
- Sequential batch process

Controllers
- Stable controllers with good tracking performance

Given
- Resolving horizon
- Job release date and due date
- Job operational stages (tasks) and capable processing units
- Sale price and fixed cost of completing a job
- Unit cost of utilities for executing a task
- Utility, safety, and quality constraints
- Dynamic models for processing tasks
- Real-time information of completed tasks and processing units

Determine
- Task assignment and sequence
- Task starting times, processing times, and processing costs
- Job completion times
- Reference trajectories for closed-loop control systems

Objective
- To maximize the production profit

We should note that a batch scheduling problem can have a great variety of configurations, for example, the sequence-dependent task changeover times and costs. However, because the focus of the article is placed on the integration of the scheduling model and the dynamic models, rather than the details about each submodel, we do not consider various scheduling configurations in this work. For example, in a batch process, the task changeover times and costs are commonly fixed parameters for the scheduling model given that the changeover procedure of transferring materials from one unit to another is typically not formulated into a dynamic model.

In this work, the integrated method is developed for the sequential batch process. We do not consider the general network-structured batch process where material splitting and mixing are allowed. The network batch process will result in a different interface between the scheduling model and the dynamic models so that a new integrated method is required, which will be considered as a future research. For simplicity, the scheduling model does not include various configurations, for example, sequence-dependent task changeover times. The controller design problem is not formulated into the integrated problem. The integrated method assumes that the controllers can stabilize the dynamic systems describing task execution and track the reference trajectories determined by the integrated method. Of course, as mentioned above, the discrepancy between the references and the real ones is allowed.

Online Integration Framework

The proposed online integration framework is shown in Figure 1d. When disruptions occur, the integrated problem is resolved to cope with the disruptions. To achieve the goal of both computational efficiency and rescheduling stability, we propose a moving horizon approach. It is a solution repairing approach under uncertainties. The moving horizon approach is based on an initial solution. Instead of changing the initial solution completely, the moving horizon approach only modifies a small portion of the initial solution. In this section, we present the framework of the online integrated method. The detailed formulation of the mathematical problem solved online will be presented in Formulation of Online Integrated Problem Section.

The horizon, in which the integrated problem is resolved, stretches from the current time point at which the information about the disruptions is received. The horizon length is equal to a predetermined value. To emphasize the difference between the integrated problem and a conventional scheduling problem, we name the horizon as the "resolving horizon" instead of the common "rescheduling horizon."

The flow chart of the moving horizon approach as well as illustrations is shown in Figure 3. Initially, the integrated scheduling and dynamic optimization problem is solved offline. The solution includes the scheduling decisions and the controller references. Then, the production is carried out according to the offline schedule and the process dynamic systems are manipulated by the controllers tracking the reference trajectories.

When disruptions occur in the process, the controllers send event messages to trigger the solution of the integrated problem online. Each event message records the information about the disruptions, for example, the changed processing times or the breakdown units. In this work, we focus on non-preemptive tasks. The controller of a task only sends the message after the task is completed or when the task is forcefully terminated by unit breakdown.

When the integrated method receives an event message, it updates the variables in the integrated problem according to the actual data. The disruptions can cause infeasibility of the initial schedule. For example, the prolonged processing time of a task makes it impossible to start the subsequent task at the scheduled starting time. The infeasibility is handled by a shifted schedule that is constructed by shifting the task bars on the Gantt to the right while the shifted distance is kept as small as possible. The shifted schedule may prolong the job completion times. In this work, no hard constraint is placed on the job completion times. Instead, a penalty function is formulated to punish the tardy jobs. Hence, shifting task bars can ensure a feasible schedule. The shifted feasible schedule serves as a good starting point from which the reduced problem is formulated and solved.

Based on the shifted schedule, the reduced problem in the rescheduling horizon is formulated. The detailed formulation is presented in the next section. The assignment and sequence of the tasks inside the rescheduling horizon can be changed as well as the recipe data. For example, the processing time of a task can be shortened to catch up the due date, whereas the processing cost is increased to achieve the shorter processing time. The solution of the reduced integrated problem also determines new controller references, which are then passed to the control systems. Eventually, the rescheduling solution is carried out and the new references are tracked by the control systems.

Because the integrated problem needs to be solved online, feasibility of the integrated problem under different conditions should be discussed. Considering that we do not place a hard constraint on the job completion times, shitting and swapping the task bars on the Gantt chart can guarantee the schedule feasibility. For the dynamic optimization, the task

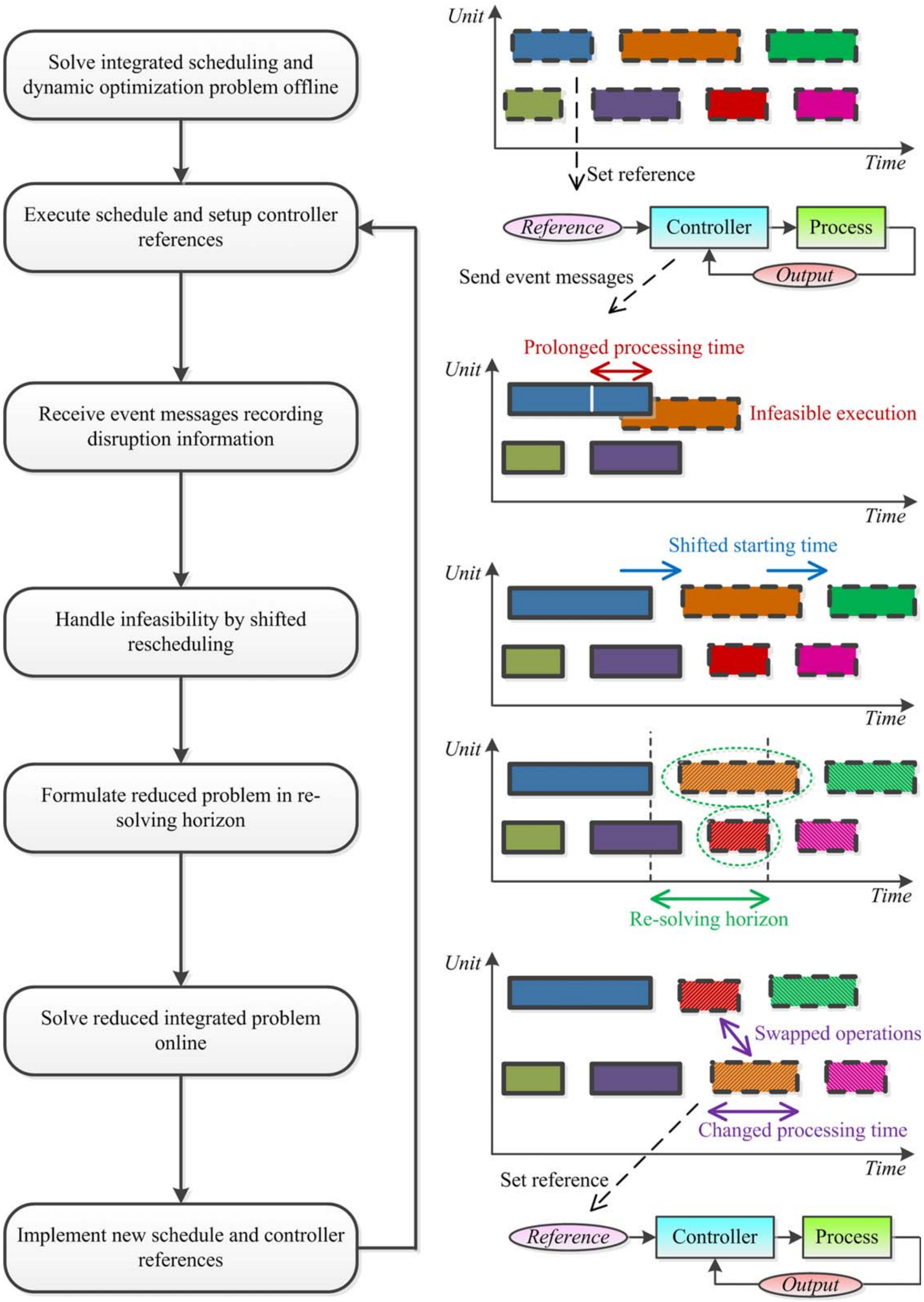


Figure 3. Flow chart of the integrated method and the illustration.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

processing times are adjustable without a hard upper bound. In addition, the initial conditions and the set-points are known, the feasibility of optimizing a dynamic model can be guaranteed. Therefore, feasibility of the online integrated problem is ensured. We should note that the integrated problem is solved to determine the reference trajectories for the control systems and it does not consider how the controllers are designed to track the determined references. According to the common nonpreemptive scheduling rule that a running task is not interrupted before its execution is completed, the integrated method does not interfere with the controllers during task processing periods. Instead, the integrated method only interacts with a controller before and after the task processing period. Therefore, we assume the controllers can

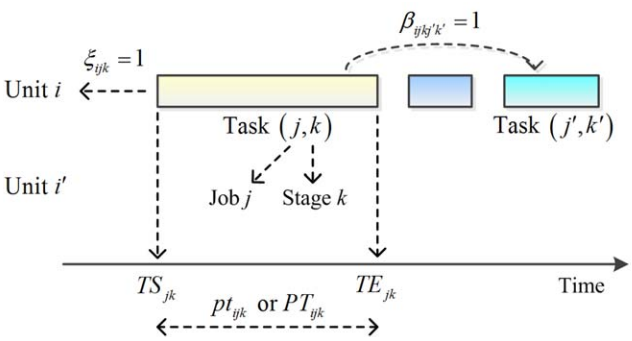


Figure 4. Formulation of the general precedence model.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

track the references determined by the integrated method. If MPCs are implemented to track the references, we assume the MPCs are recursively feasible.

Formulation of Online Integrated Problem

In this section, we present the formulation of the reduced integrated problem, which is solved online. The formulation can cover the offline integrated problem as a special case when the resolving horizon is equal to the entire production horizon.

The scheduling problem is formulated by the general precedence model,[35,36] which is illustrated in Figure 4. Let a job be indexed by j and a stage by k. All stages of job j comprise set K_j. A task, representing a job operational stage, is denoted by the pair (j,k). A processing unit is indexed by i and the units capable of executing task (j,k) belongs to set I_{jk}. The scheduling model includes two types of binary variables: the assignment variables ξ_{ijk} which indicate if task (j,k) is assigned to unit i, and the precedence variables $\beta_{ijkj'k'}$ which are equal to one if task (j,k) precedes task (j',k') in unit i. The continuous variables in the scheduling model include the task starting times TS_{jk}, the task ending times TE_{jk}. In a common scheduling problem, the task processing times are fixed as parameters. However, in the integrated problem, some tasks have variable recipe and their processing times can be changed. For the tasks with fixed recipe, the parameter processing times are denoted by pt_{ijk} while for the tasks having variable recipe, the variable processing times are denoted by PT_{ijk}.

Admittedly, there are a great variety of scheduling problems so that a single model cannot formulate all scheduling problems. Using the general precedence model,[35,36] we do not consider sequence-dependent task changeover times and costs. To formulate the sequence-dependent changeovers, the immediate precedence scheduling model is more convenient[37] where a sequencing binary variable denotes the immediate predecessor of a given task. As there is no scheduling model that outperforms the others in all scheduling problems,[37] we do not aim to find the most efficient scheduling model for formulating the integrated problem. We also should note that a scheduling specification could significantly affect the computational performance for solving the scheduling problem and in turn the performance for solving the integrated problem. For example, the sequence-dependent changeovers can substantially increase the computational time for solving the scheduling problem and consequently the one for solving the integrated problem. For scheduling problems with different specifications, the resolving horizon should be changed so that the online integrated problems can be solved within the time limit.

To present the reduced problem, we first partition the variables according to the resolving horizon. Then, we formulate the subproblems in the historical period, the resolving horizon, and the future period beyond the resolving horizon, respectively. The objective function, the cuts on the binary variables, and integrated problems are discussed in the subsequent sections.

Cluster of tasks

Let s_h denote the beginning time of the resolving horizon and l_h denote its length. In the integration framework, the resolving horizon can start at different time points when disruptions occur. However, the beginning time s_h is a parameter for each online integrated problem.

The online integrated problem is a reduced one derived from the initial solution. According to the resolving horizon, the tasks in the initial solution can be clustered into three groups as illustrated in Figure 5. Each group is represented by a set given as follows

$$\overline{\overline{JK}} = \{\text{Historical tasks} \mid \text{Start before the resolving horizon}\}$$

$$JK = \{\text{Resolving tasks} \mid \text{Start within the resolving horizon}\}$$

$$\overline{JK} = \{\text{Future tasks} \mid \text{Start after the resolving horizon}\}$$

In the reduced problem, some variables are fixed according to the historical data, whereas some are fixed according

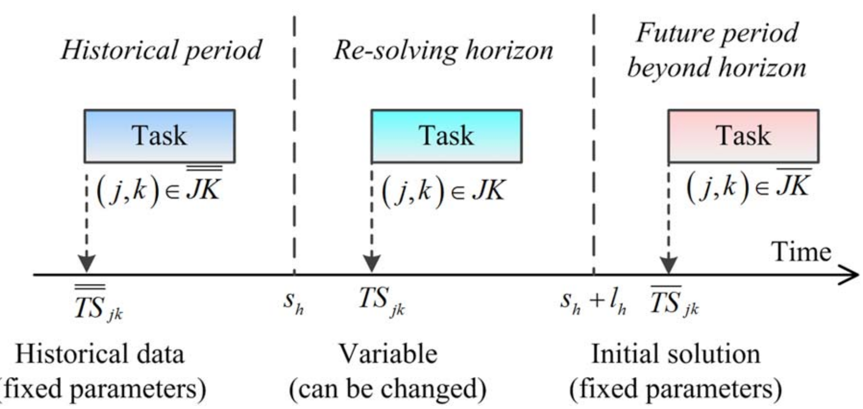


Figure 5. Clusters of tasks.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

to the initial solution. We use a variable with two bars above to represent the historical data of the variable and use a variable with one bar above to represent its value in the initial solution. For example, let TS_{jk} denote the starting time of task (j,k), Then, $\overline{\overline{\mathrm{TS}}}_{jk}$ denotes the historical data of the starting time, and $\overline{\mathrm{TS}}_{jk}$ denotes the value in the initial solution. Using the notations, TS_{jk} is a variable, whereas $\overline{\overline{\mathrm{TS}}}_{jk}$ and $\overline{\mathrm{TS}}_{jk}$ are fixed parameters. The same naming convention is applied to other variables.

Model in historical period

In the historical period, all variables are fixed according to the data from tasks that are already completed or still being executed. These tasks belong to set $\overline{\overline{JK}}$. As the assignment of the historical tasks is known, the binary assignment variables are fixed according to the historical data

$$\xi_{ijk}=\overline{\overline{\xi}}_{ijk},\ \forall (j,k)\in\overline{\overline{JK}},\ i\in I_{jk} \tag{1}$$

Similarly, the binary precedence variables $\beta_{ijkj'k'}$ are fixed at

$$\beta_{ijkj'k'}=\overline{\overline{\beta}}_{ijkj'k'},\ \forall (j,k)\in\overline{\overline{JK}},\ (j',k')\in\overline{\overline{JK}},\ i\in I_{jk}\cap I_{j'k'} \tag{2}$$

As the tasks in the historical period have all been started, the task starting times are fixed according to the historical data

$$\mathrm{TS}_{jk}=\overline{\overline{\mathrm{TS}}}_{jk},\ \forall (j,k)\in\overline{\overline{JK}} \tag{3}$$

The task ending times depend on if the tasks are completed. When the tasks are completed, the ending times are fixed according to the historical data

$$\mathrm{TE}_{jk}=\overline{\overline{\mathrm{TE}}}_{jk},\ \forall (j,k)\in\overline{\overline{JK}},\ (j,k)\ \text{is completed before}\ s_h \tag{4}$$

When the tasks are still being executed, the ending times are fixed at

$$\mathrm{TE}_{jk}=\overline{\overline{\mathrm{TS}}}_{jk}+\sum_{i\in I_{jk}}\overline{\overline{\xi}}_{ijk}\overline{\mathrm{PT}}_{ijk},\ \forall (j,k)\in\overline{\overline{JK}},\ (j,k)\ \text{is not completed before}\ s_h \tag{5}$$

where $\overline{\mathrm{PT}}_{ijk}$ is the processing time of the initial solution, representing the expected processing time.

Model in resolving horizon

In the resolving horizon, the integrated problem is solved for the tasks belonging to set JK. The integrated problem is a reduced formulation of the original entire problem. The tasks in the resolving horizon can be rescheduled and the operational recipe can be reoptimized. In a batch process, not all tasks are represented by dynamic models with control systems. For example, packing tasks are often assumed to have the fixed recipe without dynamic models.[14] To distinguish the tasks with dynamic models from the ones without, the task set is partitioned into two subsets as

$$JK=JK^{\mathrm{D}}\cup JK^{\mathrm{F}}$$

where

$$JK^{\mathrm{D}}=\{(j,k)\in JK|\text{Having dynamic model}\}$$

$$JK^{\mathrm{F}}=\{(j,k)\in JK|\text{Having fixed recipe}\}$$

A task in JK^{D} has the variable processing time PT_{ijk} and the variable processing cost PC_{ijk}, which can be manipulated by changing the controller reference. By contrast, a task belonging to JK^{F} has the fixed processing time, which is a known parameter denoted by pt_{ijk}.

Scheduling Model. All tasks in the resolving horizon can be rescheduled. The task starting times and ending times are adjustable.

For a task with the fixed recipe, the task ending time TE_{jk} is equal to the starting time TS_{jk} plus the fixed processing time pt_{ijk}

$$\mathrm{TE}_{jk}=\mathrm{TS}_{jk}+\sum_{i\in I_{jk}}\xi_{ijk}\mathrm{pt}_{ijk},\ \forall (j,k)\in JK^{\mathrm{F}} \tag{6}$$

Similarly, the ending time for a task with variable recipe is

$$\mathrm{TE}_{jk}=\mathrm{TS}_{jk}+\sum_{i\in I_{jk}}\xi_{ijk}\mathrm{PT}_{ijk},\ \forall (j,k)\in JK^{\mathrm{D}}$$

where PT_{ijk} is the variable processing time. By linearizing the product $\mathrm{XPT}_{ijk}=\xi_{ijk}\mathrm{PT}_{ijk}$, the nonlinear constraints can be reformulated into linear ones as shown below

$$\mathrm{TE}_{jk}=\mathrm{TS}_{jk}+\sum_{i\in I_{jk}}\mathrm{XPT}_{ijk},\ \forall (j,k)\in JK^{\mathrm{D}} \tag{7}$$

$$0\le \mathrm{XPT}_{ijk}\le \mathrm{PT}_{ijk},\ \forall (j,k)\in JK^{\mathrm{D}},\ i\in I_{jk} \tag{8}$$

$$\mathrm{XPT}_{ijk}\le \xi_{ijk}\mathrm{pt}_{ijk}^{\max},\ \forall (j,k)\in JK^{\mathrm{D}},\ i\in I_{jk} \tag{9}$$

$$\mathrm{XPT}_{ijk}\ge \mathrm{PT}_{ijk}-\left(1-\xi_{ijk}\right)\mathrm{pt}_{ijk}^{\max},\ \forall (j,k)\in JK^{\mathrm{D}},\ i\in I_{jk} \tag{10}$$

where $\mathrm{pt}_{ijk}^{\max}$ is an upper bound of PT_{ijk}. Because the tasks for a job are executed sequentially through the operational stages, the task in the current stage cannot start earlier than the ending time of the task in the previous stage

$$\mathrm{TS}_{jk}\ge \mathrm{TE}_{j(k-1)},\ \forall (j,k)\in JK,\ k\ge 2 \tag{11}$$

The assignment and the sequence of the tasks in the resolving horizon can be changed. The assignment variables ξ_{ijk} are constrained by

$$\sum_{i\in I_{jk}}\xi_{ijk}=1,\ \forall (j,k)\in JK \tag{12}$$

because each task is executed exactly once. In a processing unit, a task is executed either before or after another one, so the starting times of two tasks are constrained by the precedence variables β_{ijk}

$$\mathrm{TS}_{jk}+\mathrm{pt}_{ijk}-b_m\left(1-\beta_{ijkj'k'}\right)\le \mathrm{TS}_{j'k'},\ \forall (j,k)\in JK^{\mathrm{F}},\ (j',k')\in JK^{\mathrm{F}},\ (j,k)\ne(j',k'),\ i\in I_{jk}\cap I_{j'k'} \tag{13}$$

$$\mathrm{TS}_{jk}+\mathrm{PT}_{ijk}-b_m\left(1-\beta_{ijkj'k'}\right)\le \mathrm{TS}_{j'k'},\ \forall (j,k)\in JK^{\mathrm{D}},\ (j',k')\in JK^{\mathrm{D}},\ (j,k)\ne(j',k'),\ i\in I_{jk}\cap I_{j'k'} \tag{14}$$

where b_m denotes a big-M term. Inequality (13) enforces the sequence for the tasks with fixed recipe, and inequality (14) enforces the sequence for those with variable recipe. Because a unit operation is described by either dynamic model or fixed recipe, the tasks with fixed recipe and the tasks with variable recipe cannot be executed in the same unit. Thus, it is not required to formulate the sequence for a task belonging to JK^{F} and another task belonging to JK^{D}. The production sequence is only enforced when both tasks (j,k) and (j',k') are assigned to the same unit i. Thus, the precedence variables are constrained by the assignment variables

$$\xi_{ijk}+\xi_{ij'k'}-\beta_{ijkj'k'}-\beta_{ij'k'jk} \le 1,\ \forall (j,k)\in JK,\ (j',k')\in JK,\ (j,k)\ne (j',k'),\ i\in I_{jk}\cap I_{j'k'} \tag{15}$$

The inequalities ensure that if both tasks (j,k) and (j',k') are assigned to the same unit i ($\xi_{ijk}=1$ and $\xi_{ij'k'}=1$) then at least one precedence variable should be one ($\beta_{ijkj'k'}=1$ or $\beta_{ij'k'jk}=1$). The two precedence variables cannot be equal to one at the same time according to inequalities (13) and (14).

Dynamic Model. The goal of dynamic optimization in the integrated problem is to determine the reference trajectories for the controller. Only the tasks belonging to JK^{D} have dynamic models. The dynamic model of a task is described by a set of differential equations

$$\frac{dX_{ijk}(T_{ijk})}{dT_{ijk}}=F_{ijk}\left(X_{ijk}(T_{ijk}),U_{ijk}(T_{ijk})\right),\ \forall (j,k)\in JK^{\mathrm{D}},\ i\in I_{jk} \tag{16}$$

The differential equation (16) is indexed by unit i, and task (j,k). Correspondingly, all variables in the differential equations are indexed by ijk, including the time variable T_{ijk}. For a compact expression, the states $X_{ijk}(T_{ijk})$ and the inputs $U_{ijk}(T_{ijk})$ are expressed by the vector forms. The vector forms are also used to express other variables and constraints in the dynamic models.

To facilitate numerical solution of the dynamic optimization problem, the differential equation (16) can be discretized. The time interval is partitioned by equal-length segments called finite elements. The grid points are

$$\begin{aligned}&T_{ijk}^{0}=0\le T_{ijk}^{1}\le\cdots\le T_{ijk}^{r}\le\cdots\le T_{ijk}^{n_{ijk}^{r}}=\mathrm{PT}_{ijk},\ \forall (j,k)\in JK^{\mathrm{D}},\ i\in I_{jk}\\&T_{ijk}^{r}-T_{ijk}^{r-1}=L_{ijk},\ \forall (j,k)\in JK^{\mathrm{D}},\quad i\in I_{jk},\ r>0\end{aligned} \tag{17}$$

where the finite elements are indexed by r and the starting point of finite element r is denoted by T_{ijk}^{r}. The number of finite elements is denoted by n_{ijk}^{r}. The dynamic model is assumed to be time-invariant so that the initial time T_{ijk}^{0} can be set at zero. The length of the finite elements is represented by L_{ijk}. In each finite element, a number of collocation points are generated. The time value at each collocation point is

$$T_{ijk}^{rq}=T_{ijk}^{r-1}+c_qL_{ijk},\ \forall (j,k)\in JK^{\mathrm{D}},\quad i\in I_{jk},\ r>0 \tag{18}$$

where the collocation points are indexed by q. T_{ijk}^{rq} denotes the time value at collocation point q in finite element r. The coefficients c_q are determined from the discretization method.

Using the collocation points, the continuous-time state and input trajectories satisfying the differential equations are discretized. The discretization procedure transforms the differential equation (16) into algebraic equations as

$$X_{ijk}^{rq}=X_{ijk}^{r-1}+L_{ijk}\sum_{q'}a_{qq'}F_{ijk}\left(X_{ijk}^{rq'},U_{ijk}^{rq'}\right),\ \forall (j,k)\in JK^{\mathrm{D}},\ i\in I_{jk},\ r>0 \tag{19}$$

$$X_{ijk}^{r}=X_{ijk}^{r-1}+L_{ijk}\sum_{q}b_qF_{ijk}\left(X_{ijk}^{rq},U_{ijk}^{rq}\right),\ \forall (j,k)\in JK^{\mathrm{D}},\quad i\in I_{jk},\ r>0 \tag{20}$$

where X_{ijk}^{r} denotes the discretized state trajectory $X_{ijk}^{r}=X_{ijk}\left(T_{ijk}^{r}\right)$. Similarly X_{ijk}^{rq} denotes the discretized state value at a collocation point and U_{ijk}^{rq} is the discretized value of the input trajectory. The parameter $a_{qq'}$ represents a coefficient in the collocation matrix and b_q is an element in the collocation vector. The discretization procedure is actually a numerical integration method for solving the differential equations. Various methods with different collocation points can be applied. In this work, we use the Radau IIA method, which has a robust performance for stiff differential equations.[38]

The initial condition and the final value of the dynamic models are

$$X_{ijk}^{(r=0)}=x_{ijk}^{I},\ \forall (j,k)\in JK^{\mathrm{D}},\quad i\in I_{jk} \tag{21}$$

$$X_{ijk}^{\left(r=n_{ijk}^{r}\right)}=x_{ijk}^{F},\ \forall (j,k)\in JK^{\mathrm{D}},\quad i\in I_{jk} \tag{22}$$

where the initial condition x_{ijk}^{I} and the final value x_{ijk}^{F} are given parameters. Besides, the dynamic models are subject to constraints regarding safety, quality, and other issues. These constraints can be summarized as

$$H_{ijk}\left(X_{ijk}^{rq},U_{ijk}^{rq}\right)\le 0,\ \forall (j,k)\in JK^{\mathrm{D}},\quad i\in I_{jk} \tag{23}$$

Optimizing the dynamic models returns processing times and processing costs which are recipe data for the scheduling model. Without loss of generality, the processing cost can be represented as a function φ_{ijk} of the final state value

$$PC_{ijk}=\varphi_{ijk}\left(X_{ijk}^{\left(r=n_{ijk}^{r}\right)}\right),\ \forall (j,k)\in JK^{D},\quad i\in I_{jk} \tag{24}$$

The processing time is equal to the product

$$PT_{ijk}=L_{ijk}n_{ijk}^{r} \tag{25}$$

Model in future period beyond resolving horizon

For the future tasks beyond the resolving horizon, the binary variables and the processing times are fixed at those determined by the initial solution. The starting times can be varied to guarantee the feasibility, whereas the processing times are fixed according to the initial solution.

The tasks in this period belong to set $\overline{JK}$. The assignment variables are fixed at

$$\xi_{ijk}=\overline{\xi}_{ijk},\ \forall (j,k)\in \overline{JK},\ i\in I_{jk} \tag{26}$$

where $\overline{\xi}_{ijk}$ is the value of the binary variable in the initial solution. The precedence variables are fixed at the initial solution value $\overline{\beta}_{ijkj'k'}$

$$\beta_{ijkj'k'}=\overline{\beta}_{ijkj'k'},\ \forall (j,k)\in\overline{JK},\ (j',k')\in\overline{JK},\ i\in I_{jk}\cap I_{j'k'} \tag{27}$$

The task processing times are fixed at

$$\mathrm{PT}_{ijk} = \overline{\mathrm{PT}}_{ijk},\ \ \forall (j,k) \in \overline{JK},\ \ i \in I_{jk} \tag{28}$$

The task starting times TS_{jk} and the ending times TE_{jk} are variables, which are constrained by

$$\mathrm{TE}_{jk} = \mathrm{TS}_{jk} + \sum_{i \in I_{jk}} \overline{\xi}_{ijk}\mathrm{pt}_{ijk},\ \ \forall (j,k) \in \overline{JK}^{\mathrm{F}} \tag{29}$$

$$\mathrm{TE}_{jk} = \mathrm{TS}_{jk} + \sum_{i \in I_{jk}} \overline{\xi}_{ijk}\overline{\mathrm{PT}}_{ijk},\ \ \forall (j,k) \in \overline{JK}^{\mathrm{D}} \tag{30}$$

where set $\overline{JK}^{\mathrm{F}}$ indicates the tasks with the fixed recipe and set $\overline{JK}^{\mathrm{D}}$ includes those with the variable recipe. The two sets partition $\overline{JK}$, that is, $\overline{JK}^{\mathrm{F}} \cup \overline{JK}^{\mathrm{D}} = \overline{JK}$.

Objective function

The schedule performance often relies on the job completion times. It is desirable to deliver the completed jobs as soon as possible. For the adjustable processing times, small values are preferable to reduce the job completion times. However, short processing times may incur large processing costs. There should be a tradeoff between the processing times and the processing costs.

The objective function for the integrated problem is to maximize the profit,[39] which is the difference between the sales and the costs

$$\max\ \ \mathrm{Profit} = \mathrm{Sales} - \mathrm{Cost}^{\mathrm{V}} - c^{\mathrm{F}} \tag{31}$$

The sales is equal to the sum of the job prices

$$\mathrm{Sales} = \sum_{j} \mathrm{PR}_{j} \tag{32}$$

The variable cost is the sum of processing costs for the tasks with variable recipe

$$\mathrm{Cost}^{\mathrm{V}} = \sum_{(j,k)\in JK^{D},\ i\in I_{jk}} \xi_{ijk}\mathrm{PC}_{ijk}$$

The bilinear terms can be linearized by introducing continuous variables $\mathrm{XPC}_{ijk} = \xi_{ijk}\mathrm{PC}_{ijk}$ such that

$$\mathrm{Cost}^{\mathrm{V}} = \sum_{(j,k)\in JK^{D}, i\in I_{jk}} XPC_{ijk} \tag{33}$$

$$0 \le \mathrm{XPC}_{ijk} \le \mathrm{PC}_{ijk},\ \ \forall (j,k) \in JK^{\mathrm{D}},\ \ i \in I_{jk} \tag{34}$$

$$\mathrm{XPC}_{ijk} \le \xi_{ijk}\mathrm{pc}_{ijk}^{\max},\ \ \forall (j,k) \in JK^{\mathrm{D}},\ \ i \in I_{jk} \tag{35}$$

$$\mathrm{XPC}_{ijk} \ge \mathrm{PC}_{ijk} - \left(1-\xi_{ijk}\right)\mathrm{pc}_{ijk}^{\max},\ \ \forall (j,k) \in JK^{\mathrm{D}},\ \ i \in I_{jk} \tag{36}$$

The processing cost is bounded from above by $\mathrm{pc}_{ijk}^{\max}$. The fixed cost c^{F} is a parameter, including other unchanged costs.

The job prices PR_j are functions on the job completion times DT_j as shown in Figure 6. There is a threshold d_j on each completion time. When the completion time is less than the threshold value, the job has a constant price pr_j.

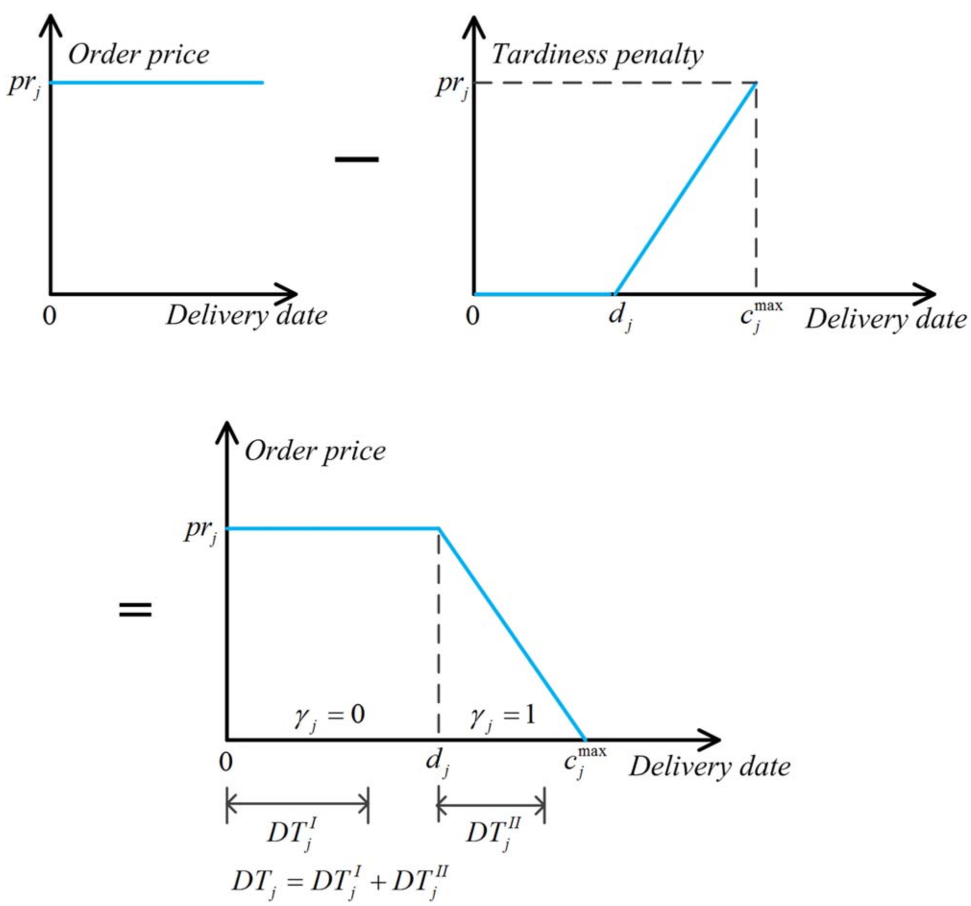


Figure 6. Job price function.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

However, when the completion time exceeds the threshold, a linear penalty term is added and the job price reduces gradually as the completion time increases. The completion time cannot exceed the upper bound $c_j^{\max}$.

The penalty on the completion time results in a piecewise linear job price function, which has two segments. Binary variables γ_j are introduced to indicate if a job completion time exceeds the threshold ($\gamma_j = 1$) or not ($\gamma_j = 0$). Then, the job completion times can be expressed by

$$\mathrm{DT}_j = \mathrm{DT}_j^{\mathrm{I}} + \mathrm{DT}_j^{\mathrm{II}}, \ \forall j \tag{37}$$

$$\gamma_j d_j \leq \mathrm{DT}_j^{\mathrm{I}} \leq d_j, \ \forall j \tag{38}$$

$$0 \leq \mathrm{DT}_j^{\mathrm{II}} \leq \gamma_j \left(c_j^{\max} - d_j \right), \ \forall j \tag{39}$$

where $\mathrm{DT}_j^{\mathrm{I}}$ and $\mathrm{DT}_j^{\mathrm{II}}$ are the components of the completion time DT_j. The job completion time is equal to the ending time of the last task

$$\mathrm{DT}_j = \mathrm{TE}_{j(k=|K_j|)}, \ \forall j \tag{40}$$

The completion times are constrained by the upper bound

$$\mathrm{DT}_j \leq c_j^{\max}, \ \forall j \tag{41}$$

The job price is equal to

$$\mathrm{PR}_j = \mathrm{pr}_j \left(1 - \frac{1}{c_j^{\max} - d_j} \mathrm{DT}_j^{\mathrm{II}} \right), \ \forall j \tag{42}$$

Cuts on binary variables

As an MINLP, the integrated problem is complicated by coupling of binary variables and nonlinear equations. To simplify the solution approach, some constraints on the binary variables are introduced, which cutoff some infeasible region and help the solver to reduce the search scope.

Some binary variables can be fixed at zero for the infeasible assignment

$$\xi_{ijk} = 0, \ \forall j, \ \left(k \notin K_j, \ \text{or} \ i \notin I_{jk} \right) \tag{43}$$

and the infeasible precedence

$$\beta_{ijkj'k'} = 0, \ \forall j, \ \ j', \ \left(k \notin K_j, \ \text{or} \ k' \notin K_{j'}, \ \text{or} \ (j,k) = (j',k'), \ \text{or} \ i \notin I_{jk} \cap I_{j'k'} \right) \tag{44}$$

Task clustering according to the resolving horizon also adds cuts on the precedence constraints. The tasks belonging to the same period can be fixed according to Eqs. (2) or (27). For the tasks belonging to different periods, there are also constraints on the precedence variables. As the historical tasks precede the tasks in the resolving horizon, the precedence variables are

$$\beta_{ijkj'k'} = \bar{\bar{\xi}}_{ijk} \xi_{ij'k'}, \ \forall (j,k) \in \overline{\overline{JK}}, \ (j',k') \in JK, \ i \in I_{jk} \cap I_{j'k'} \tag{45}$$

The value of $\beta_{ijkj'k'}$ is one if tasks (j,k) and (j',k') are assigned to the same processing unit. The precedence variables for the historical tasks and the future tasks are

$$\beta_{ijkj'k'} = \bar{\bar{\xi}}_{ij'k'} \bar{\xi}_{ijk}, \ \forall (j,k) \in \overline{\overline{JK}}, \ (j',k') \in \overline{JK}, \ i \in I_{jk} \cap I_{j'k'} \tag{46}$$

which are known parameters. Conversely, no tasks in the resolving horizon or the future period can precede the historical tasks

$$\beta_{ijkj'k'} = 0, \ \forall (j,k) \notin \overline{\overline{JK}}, \ (j',k') \in \overline{\overline{JK}}, \ i \in I_{jk} \cap I_{j'k'} \tag{47}$$

The similar constraints can be formulated for the tasks in the resolving horizon and the tasks beyond the resolving horizon

$$\beta_{ijkj'k'} = \xi_{ijk} \bar{\xi}_{ij'k'}, \ \forall (j,k) \in JK, \ (j',k') \in \overline{JK}, \ i \in I_{jk} \cap I_{j'k'} \tag{48}$$

$$\beta_{ijkj'k'} = 0, \ \forall (j,k) \in \overline{\mathrm{JK}}, \ (j',k') \in JK, \ i \in I_{jk} \cap I_{j'k'} \tag{49}$$

Integrated problems

To summarize, the online reduced integrated problem is formulated as

(Online Integration)	max Profit s.t. Scheduling model (1)–(15), (26)–(49) Dynamic models (17)–(25)

Besides the online reduced integrated problem, we need to solve the entire problem offline to start the production process. We also need to determine the shifted schedule which generates a feasible initial solution starting from which the online problem is derived. These two problems can be regarded as the special cases of the online problem (Online Integration).

The offline problem is formulated by setting the starting time as zero and the length of the resolving horizon as the production horizon. As a result, the task sets become

$$\overline{\overline{JK}} = \overline{JK} = \{\}$$

$$JK = \{\text{All tasks}\}$$

The offline integrated problem is

(Offline Integration)	max Profit s.t. Scheduling model (6)–(15), (31)–(44) Dynamic models (17)–(25)

The shifted schedule can be obtained by setting the length of the resolving horizon as zero. Then, the task set in the resolving horizon is empty

$$\overline{\overline{JK}} \cup \overline{JK} = \{\text{All tasks}\}$$

$$JK = \{\}$$

All variables are fixed except the starting times of the tasks belonging to $\overline{JK}$. To minimize the change in the starting times, a penalty term is formulated as

$$\mathrm{VST} = \sum_{(j,k) \in \overline{JK}} \left| \mathrm{TS}_{jk} - \overline{\mathrm{TS}}_{jk} \right|$$

The absolute difference can be expressed by

$$\mathrm{VST} = \sum_{(j,k)\in\overline{JK}} \mathrm{DS}_{jk} \tag{50}$$

$$\mathrm{DS}_{jk} \geq \mathrm{TS}_{jk} - \overline{\mathrm{TS}}_{jk}, \ \forall (j,k) \in \overline{JK} \tag{51}$$

$$\mathrm{DS}_{jk} \geq \overline{\mathrm{TS}}_{jk} - \mathrm{TS}_{jk}, \ \forall (j,k) \in \overline{JK} \tag{52}$$

where $\mathrm{DS}_{jk} = |\mathrm{TS}_{jk} - \overline{\mathrm{TS}}_{jk}|$. Either inequality (51) or inequality (52) will be activated when the penalty term is added to the objective function. The shifted rescheduling problem is

(Shifted schedule)	max Profit—VST s.t. Scheduling model (1)–(5), (26)–(44), (46), (47), (50)–(52)

Case Study

To demonstrate the proposed online integrated method, we apply it to a simulated example in this section. The results of the integrated method are first compared with those returned by the sequential method. The integrated method can significantly improve the objective function value meanwhile the required computational time increases substantially. The computational complexity makes it very difficult to solve the entire integrated problem online. The advantage of the proposed method is that it only solves a reduced problem in the resolving horizon, ensuring computational efficiency as well as rescheduling stability.

We should note that the online solution approach is different from the conventional rescheduling ones. In the case study, though the scheduling problem is not large-scale, the integrated problem, a complicated MIDO, is actually difficult to solve. The complexity of the case study is comparable to those in the literature.[13,14,39] The integrated problem is so complicated that it cannot be solved to the 1% optimality gap within 50,000 s.

In this work, all optimization problems are modeled in GAMS 24.0.1[40] and solved in a PC with Intel(R) Core(TM) i5-2400 CPU @ 3.10 GHz, 8 GB RAM, and Window 7 64-bit operating system.

Scheduling model

The process diagram is shown in Figure 7. The batch process consists of three operational stages: a reaction task, a filtration task, and another reaction task. The first reaction task can be executed in Reactor R_{I} or R_{II} and the second reaction task can be executed in Reactor R_{III} or R_{IV}. The filtration task is processed in a filter F. The batch process aims to complete eight jobs. The filtration stage of a job has the fixed processing time and the processing cost. The two reaction stages of a job are described by dynamic models, which will be given in Dynamic model Section. The objective is to maximize the production profit. The job due date is 15 h and the value for completing a job will decrease when the job completion time exceeds the due date. The parameter values for the scheduling model are listed in Table A1.

Dynamic model

The dynamic model for a reaction task is described by the differential equations

Chemical kinetics $S \rightarrow P$

$$\frac{dC_{\mathrm{S}}(t)}{dt} = -k_{\mathrm{S}} e^{-e_{\mathrm{S}}/T_{\mathrm{R}}(t)} C_{\mathrm{S}}(t)$$

$$\frac{dI_T(t)}{dt} = T_{\mathrm{R}}(t) - t_{\mathrm{R}}^{\min}$$

$$C_{\mathrm{S}}(0) = c_{\mathrm{S}}^0$$

$$I_T(0) = 0$$

$$0 \leq t \leq \mathrm{PT}$$

$$\mathrm{PC} = c^{\mathrm{R}} I_T(\mathrm{PT})$$

$$t_{\mathrm{R}}^{\min} \leq T_{\mathrm{R}}(t) \leq t_{\mathrm{R}}^{\max}$$

$$C_{\mathrm{S}}(\mathrm{PT}) \leq c_{\mathrm{S}}^0 \chi_{\mathrm{S}}$$

$$T_{\mathrm{R}}(\mathrm{PT}) \leq t_{\mathrm{R}}^{\mathrm{final}}$$

The state variable C_{S} denotes the reactant concentration. The reaction follows the first-order mass reaction law and

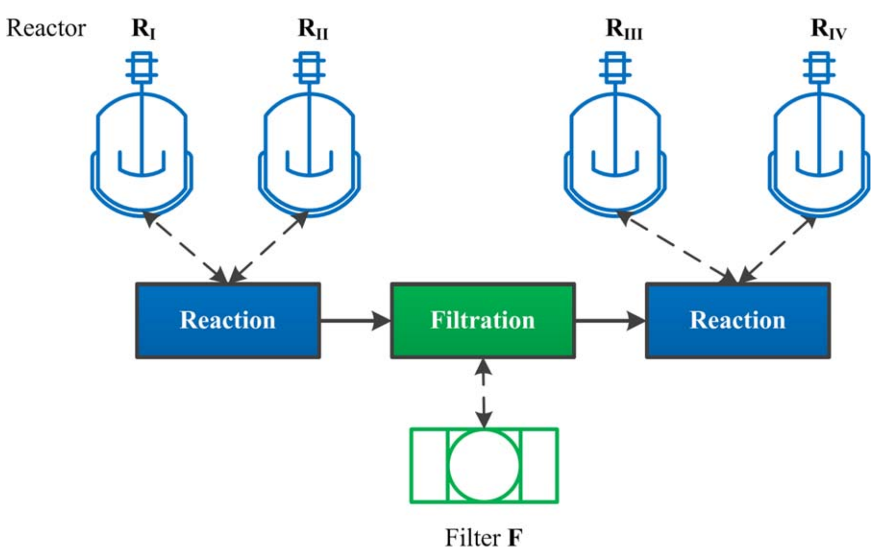


Figure 7. Process diagram for the case study.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the rate coefficient is dependent on the reactor temperature T_R. The time interval of a dynamic model is bounded by the processing time PT. The processing cost PC is proportional to the final value of the integral temperature I_T. The differential equation of I_T implies

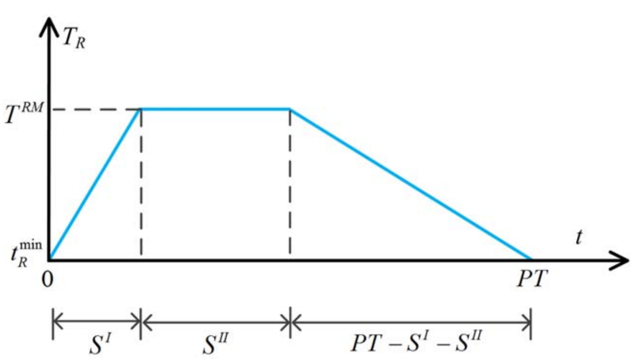
$$\text{PC} = c^{\text{R}} \int_{0}^{\text{PT}} \left(T_{\text{R}}(t) - t_{\text{R}}^{\min}\right) dt$$

The coefficient c^{R} denotes the unit utility cost. The calculation of the processing cost follows that in the literature.[14]

For the safety issue, the reactor temperature is bounded by the minimum value $t_{\text{R}}^{\min}$ and the maximum value $t_{\text{R}}^{\max}$. The final value of the concentration should satisfy the demanded conversion χ_S and the final value of the reactor temperature should be less than a given value $t_{\text{R}}^{\text{final}}$ so that the outlet materials can be cool enough for the safe transmission to a storage tank. The parameters of the reaction dynamic models are listed in Table A2.

The total number of dynamic models in the integrated problem is $8 \times 2 \times 2 = 32$ (#jobs $\times$ #reaction tasks $\times$ #units). The differential equations of each dynamic model are discretized by the collocation method using 30 finite elements. The number of finite elements is determined by a trial-and-error method on all dynamic models. Further increasing the number of finite elements does not change the results of dynamic optimization. Though a single dynamic model is simple, the 32 dynamic models can generate a great number of nonlinear equations after the discretization procedure.

The reactor temperature is the controller reference, which will be tracked by the temperature control system. The temperature reference trajectory is often represented by a trapezoid,[41] which is shown in Figure 8. The value of T^{RM}, S_{I}, S_{II}, and PT will be determined by solving the integrated problem. To cope with the piecewise linear function of the temperature reference, we use the multiperiod dynamic optimization approach. The dynamic optimization problem is solved in three periods corresponding to the three segments of the temperature reference. In each period, the temperature is represented by a linear function of the time. The results of different periods are linked in such a way that the final value of the dynamic model in the previous period is the initial condition of the current period.



$$T_R = \begin{cases} t_R^{\min} + \dfrac{T^{RM}}{S^I} t, & 0 \le t < S^I \\ T^{RM}, & S^I \le t \le S^I + S^{II} \\ t_R^{\min} + \dfrac{T^{RM}}{PT - S^I - S^{II}} (PT - t), & S^I + S^{II} < t \le PT \end{cases}$$

Figure 8. Reference trajectory of the reactor temperature.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Offline schedule by sequential method

For the comparison purpose, we first solve the integrated problem by the conventional sequential method. The advantage of the sequential method is its simplicity. The sequential method determines the processing times and the processing costs by solving a set of separated dynamic optimization problems for each dynamic model. The processing times and the processing costs are then fixed as parameters when the scheduling problem is solved. The sequential method optimizes each dynamic model by minimizing the processing time. A small processing time can reduce the completion time of a job and in turn increase the sales. However, a small processing time may imply a large processing cost. The inability to tradeoff the conflicting factors of the processing times and the processing costs is the main disadvantage of the sequential method.

After the discretization procedure, each dynamic model contains 548 equations and 551 variables. The NLP solver CONOPT is used to solve the dynamic optimization problem. It spends 10.6 s in total to minimize the processing times of the 32 dynamic models. The scheduling model contains 1227 equations, 89 continuous variables, and 314 binary variables. The MILP solver CPLEX 12.5 spends 0.8 s to solve the scheduling problem to the 1% optimality gap. The optimal profit returned by the sequential method is 236.3 m.u. (monetary unit). The Gantt chart of the sequential method is shown in Figure 9.

Offline schedule by integrated method

To optimize the overall performance of the batch plant, the integrated method is required. The integrated problem (Offline Integration) is solved by the simultaneous approach.[9,13,14] The integrated problem is initialized by the results of the sequential method. Though each subproblem is easy to solve, the integrated problem is very complicated. The model includes 18,892 equations, 17,722 continuous variables, and 314 binary variables. The integrated problem is solved by SBB (We have

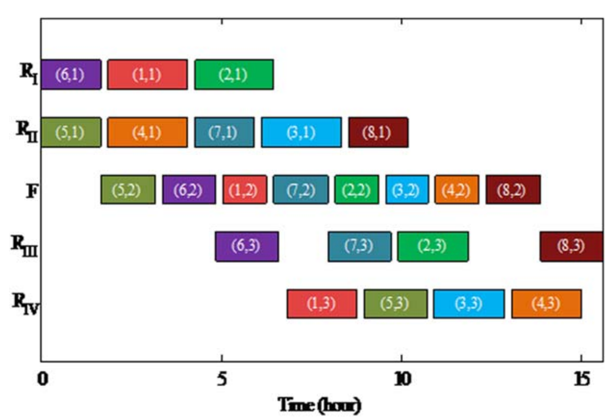


Figure 9. Scheduling results returned by the sequential method. The label on a task bar denotes (job, stage).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

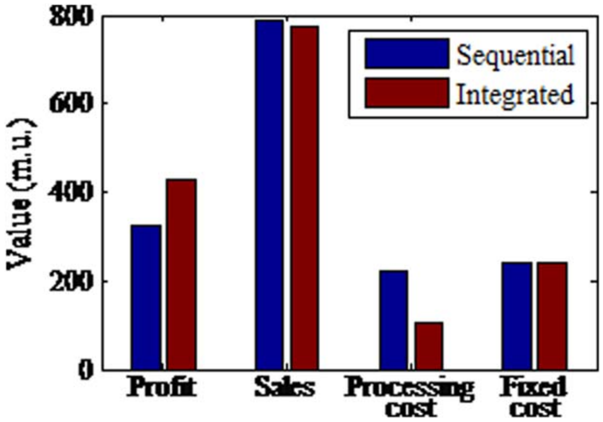


Figure 10. Components of the objective function value returned by the sequential method and the integrated method.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

also tried another common MINLP solver DICOPT, which had difficulty in finding a feasible solution because of a large number of nonlinear equations in the integrated problem). When the resource limit of 50,000 s (13.9 h) is reached, the optimality gap remains as large as 17.6%. Such inefficiency of the simultaneous method is also reported in previous studies,[13,14] reflecting the complexity of the integrated problem.

Though the optimality gap is large, the best solution found by the integrated method is 429.4 m.u. Compared with 326.3 m.u. returned by the sequential method, a 31.6% improvement is achieved by the integrated method. To investigate the tradeoff made by the integrated method, the components of the objective function value are listed in Figure 10. The sequential method attempts to increase the sales by minimizing the task processing times. However, the minimum processing times incur large processing costs, lowering the profit. The integrated method can make a better balance between the sales and the total processing cost. Though the sale is slightly lower than that of the sequential method, the total processing cost is substantially reduced, leading to a much higher profit.

The Gantt chart of the integrated method is shown in Figure 11. By changing the processing times simultaneously with the production sequence and the unit assignment, the integrated method can utilize the processing units more effectively. The idle periods of the four reactors in Figure 11 are much shorter than those by the sequential method in Figure 9.

We should note that the advantage of the integrated method relies on the fact that it can make a better tradeoff on the conflicting factors than its sequential counterpart. The conflicting factors in the case study are the processing times and the processing costs when a processing cost increases as its corresponding processing time decreases. Otherwise, no tradeoff is required when the processing cost increases with the processing time so the integrated method is not need. In this case, both sequential and integrated methods return the same result.

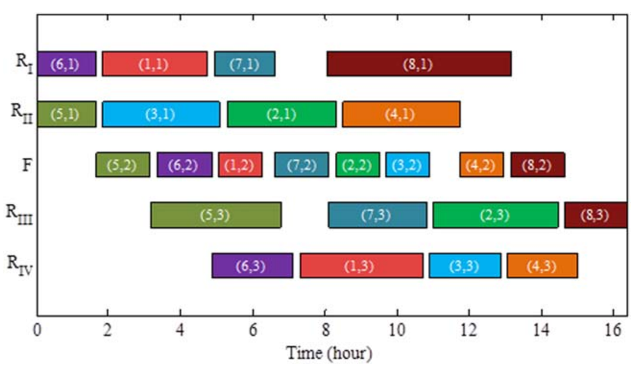


Figure 11. Scheduling results returned by the integrated method.

The label on a task bar denotes (job, stage). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Track of temperature references by control systems

In the proposed integration framework, the integrated problem is solved to determine the controller reference. In this subsection, we investigate the tracking ability of control systems. It will be shown that though there are discrepancy between the reference temperature and the actual one, the discrepancy is negligible. Even when some minor disruptions occur in the control system, the controller can automatically attenuate them. There is no need to resolve the integrated problem for any disruption. The integrated problem is only solved online when some major disruptions occur, which cannot be handled by the control system.

The temperature control system is shown in Figure 12. The temperature reference $T_R(t)$ is given by solving the integrated problem. The controller measures the actual reactor temperature $T_R^m(t)$ and manipulates the flow rates of the heating fluid and the cooling fluid. The flow rates change the jacket temperature $T_J(t)$ which in turn changes the reactor temperature $T_R^m(t)$ through heat exchange.

MPC is implemented for the temperature control and it determines the jacket flow rates by solving the optimization problem

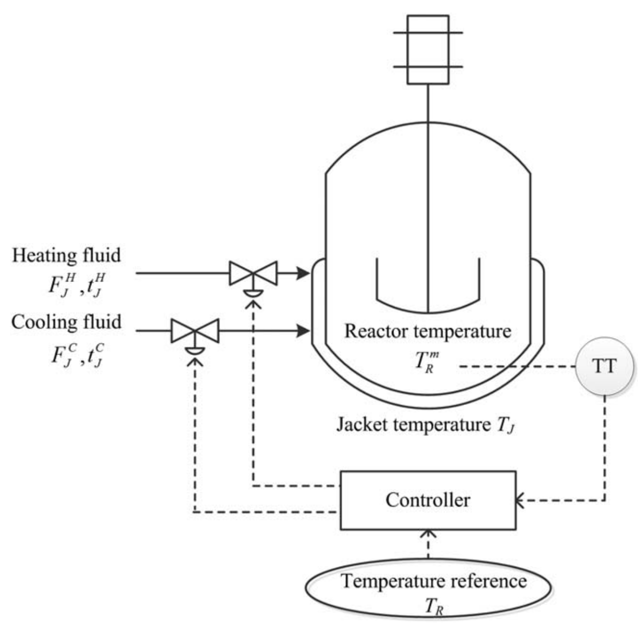


Figure 12. Temperature control loop of a reactor.

$$F_{\rm J}^{\rm H}(t), F_{\rm J}^{\rm C}(t) = \arg\min \int_0^{s_{\rm TC}} \left(T_{\rm R}^{\rm m}(t) - T_{\rm R}(t)\right)^2 dt$$

s.t.

$$\frac{dT_{\rm R}^{\rm m}(t)}{dt} = \frac{ua_{\rm J}}{v_{\rm R}\rho_{\rm R}cp_{\rm R}}\left(T_{\rm J}(t) - T_{\rm R}^{\rm m}(t)\right)$$

$$\frac{dT_{\rm J}(t)}{dt} = \frac{F_{\rm J}^{\rm H}(t)\left(t_{\rm J}^{\rm H} - T_{\rm J}(t)\right)}{v_{\rm J}} + \frac{F_{\rm J}^{\rm C}(t)\left(t_{\rm J}^{\rm C} - T_{\rm J}(t)\right)}{v_{\rm J}} + \frac{ua_{\rm J}}{v_{\rm J}\rho_{\rm J}cp_{\rm J}}\left(T_{\rm R}^{\rm m}(t) - T_{\rm J}(t)\right)$$

$$t_{\rm R}^{\rm min} \le T_{\rm R}^{m}(t) \le t_{\rm R}^{\rm max}$$

$$f_{\rm JH}^{\rm min} \le F_{\rm J}^{H}(t) \le f_{\rm JH}^{\rm max}$$

$$f_{\rm JC}^{\rm min} \le F_{\rm J}^{\rm C}(t) \le f_{\rm JC}^{\rm max}$$

The horizon $s_{\rm TC}$ for MPC is set as 1 h. The objective is to minimize the discrepancy between the reactor temperature $T_{\rm R}^{\rm m}(t)$ and the reference value $T_{\rm R}(t)$. The state variables are the reactor temperature $T_{\rm R}^{\rm m}(t)$ and the jacket temperature $T_{\rm J}(t)$. The two differential equations describe the energy exchange between the reactor and the jacket. The control variables are the flow rates of the heating fluid $F_{\rm J}^{\rm H}(t)$ and the cooling fluid $F_{\rm J}^{\rm C}(t)$. The reactor temperature and the flow rates are bounded by the path constraints. All batch reactors in the case study are identical whose parameters are shown in Table A3. The controlled flow rates are set as piecewise constant functions. The tracking results of the controller are shown in Figure 13. It is seen that the actual reactor temperature can closely track the reference value.

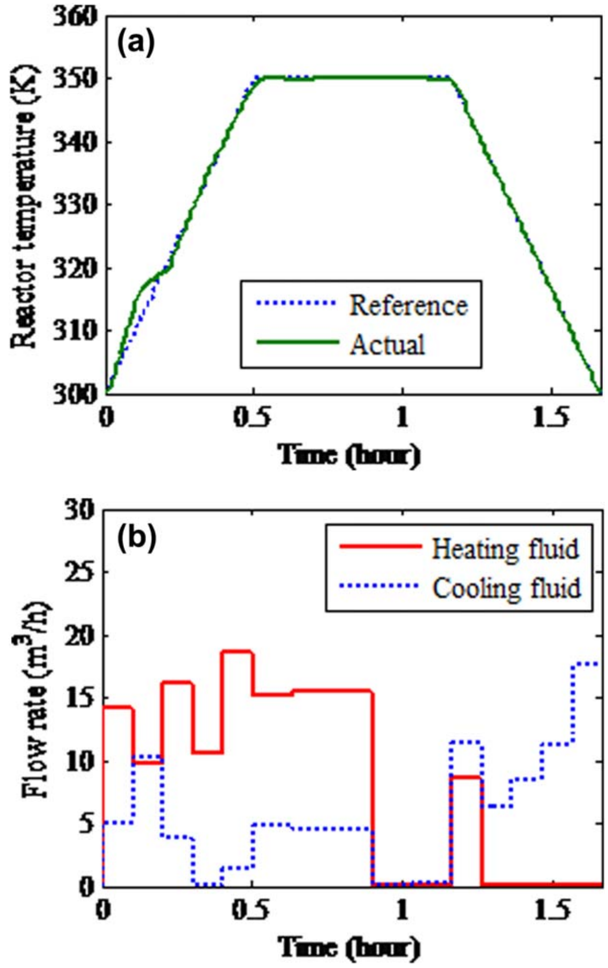


Figure 13. Trajectories of the control system for task (6,1).

(a) Temperature trajectories. (b) Flow rate trajectories. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Online implementation of integrated method

When uncertainties occur, the information concerning disruptions will be passed to the integrated problem. Then, the integrated problem is resolved online. In this case study, the time available for the integrated method is set as 0.1 h (360 s). The time period is not enough to solve the entire integrated problem as the computational time for the offline solution is nearly as long as the production horizon. The proposed method can, however, return a solution in a short time period because it merely solves a reduced problem.

To demonstrate the proposed method, we create two uncertainty scenarios: one is caused by the control system disruption and the other by the unit breakdown. To demonstrate the necessity of resolving the integrated problem online, the results are compared with those returned by the shifted schedule. The shifted schedule only changes the task starting times, whereas it does not change the task processing times and processing costs by reoptimizing the dynamic systems.

Scenario 1: Control System Disruption. In this scenario, a large disruption occurs in the control system for task (6,1). The control system fails to work in the third segment for the temperature reference. The heating flow rate remains at zero, whereas the cooling flow rate is fixed at 8 m^3/h. Though the actual temperature can finally decrease to the set value, the total processing time is prolonged by the disruption. The prolonged processing time makes the original schedule infeasible. The next task in the reactor cannot start according to the offline schedule.

The proposed method first generates a feasible schedule by solving the shifted rescheduling problem (Shifted schedule). The shifted schedule is shown in Figure 14a. The shifted schedule determines the resolving horizon starting from the completion time of the task (6,1) with the length equal to 5 h.

Then, the tasks are clustered into three groups according to the resolving horizon. The online integrated problem (Online Integration) is solved. Only the tasks in the resolving horizon are free to change. They can be resequenced or reassigned. The starting times and the processing times can be changed as well. The reference for the reactor temperature can be reoptimized. The results of the online solution are shown in Figure 14b.

The online solution of the reduced integrated problem is different from the shifted schedule in terms of task sequence and assignment. For example, the online solution assigns reactor $R_{\rm III}$ to execute task (6,3) instead of reactor $R_{\rm IV}$ for the shifted schedule. We should note that the integrated problem is solved not only to change scheduling decisions but also to change the recipe data of processing times and processing costs. It is seen from the Gantt charts that the processing time for task (6,3) is prolonged in the online solution. The changed recipe data correspond to the different controller references which are shown in Figure 15. In contrast, the shifted schedule fixes the controller references at

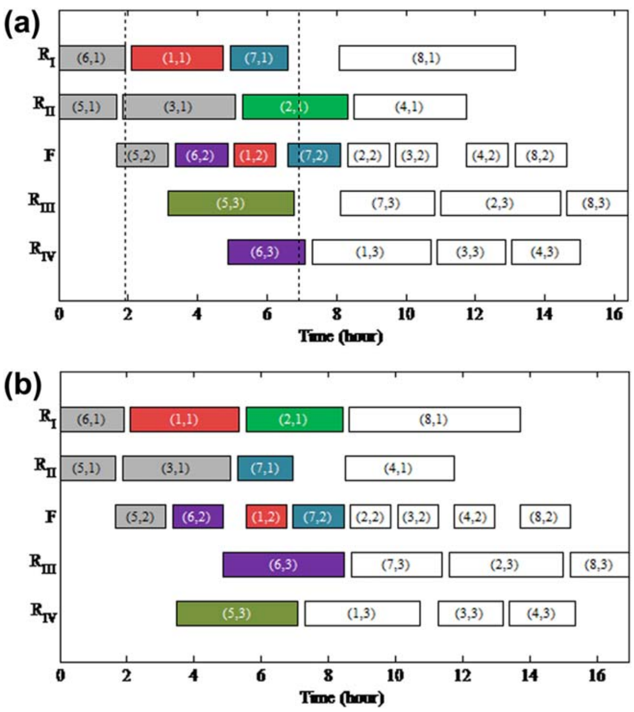


Figure 14. The scheduling results by resolving the integrated problem under uncertainty scenario 1.

(a) Shifted schedule. (b) Online solution. The label on a task bar denotes (job, stage). The processing time of task (6,1) is prolonged by the disruption. The period between two dash lines in the shifted schedule is the resolving horizon. The historical tasks are represented in gray and the future tasks beyond the resolving horizon are represented by white boxes. The tasks in the resolving horizon are colored. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

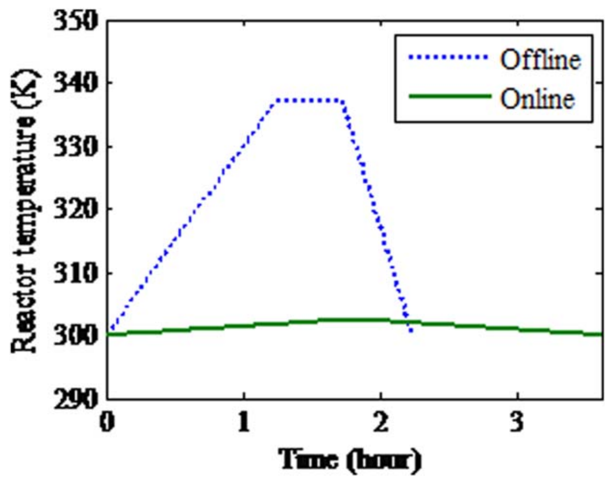


Figure 15. The controller references for task (6,3) when the integrated problem is solved offline and when the problem is solved online under uncertainty scenario 1.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the values in the offline solution. Rescheduling the production along with changing the operational conditions is a feature of the integrated method, which is distinct from a common rescheduling approach.

The comparisons of the objective function values for the shifted schedule and the online solution are listed in Table 1. The reduced integrated problem includes 7456 equations, 7018 continuous variables, and 126 binary variables. Compared with the offline integrated problem, the online problem is reduced so that it can be solved to the 1% optimality gap in 99.4 s, which is less than the limit of 360 s. Under the uncertainty, the profit is degenerated from 429.4 m.u. of the offline solution to 417.0 m.u. of the shifted schedule. The profit is degenerated by 3% for the shifted schedule, whereas it is degenerated by 1% for the online solution.

Scenario 2: Processing Unit Breakdown. In this scenario, the disruption of unit breakdown is investigated. Unit breakdown is a common uncertainty in scheduling problem. It cannot be coped with by a local control system and the integrated problem has to be solved online. In this example, the process is initially operated according to the offline solution in Figure 11. Reactor R_{II} breaks down from 0 to 1 h. The breakdown delays the starting time of the first task (5,1) in the reactor. When the reactor is recovered at 1 h, the integrated problem is solved by the proposed method.

Table 1. Comparisons of the Shifted Schedule and the Online Solution of the Integrated Problem Under Uncertainty Scenario 1

	Shifted Schedule	Online Solution
Profit (m.u.)	417.0	425.2
Sales (m.u.)	764.1	754.3
Processing cost (m.u.)	107.1	89.1
Fixed cost (m.u.)	240	240

First, the shifted rescheduling problem (Shifted schedule) is solved. The shifted schedule is shown in Figure 16a. The resolving horizon extends from 1 to 6 h. Then, the online integrated problem (Online Integration) is solved and the solution is shown in Figure 16b. The online solution changes the task sequence and assignment. For example, the task (7,1) changes to the second position from the third position in the shifted schedule. The operational condition is also changed. The processing time of task (7,1) is prolonged to reduce the processing cost.

The comparisons of the objective function values for the shifted schedule and the online solution are listed in Table 2. The reduced integrated problem includes 8942 equations, 8426 continuous variables, and 138 binary variables. The online problem is reduced so that it can be solved to the 1% optimality gap in 151.3 s, which is less than the limit of 360 s. Under the uncertainty, the profit is degenerated from 429.4 m.u. of the offline schedule to 377.6 m.u. of the shifted schedule. The profit is degenerated by 12.1% for the shifted schedule, whereas it is degenerated only by 1.5% for the online solution.

Conclusions

We propose an online integrated method for the sequential batch process. The integrated method determines the controller references simultaneously with scheduling decisions and task recipes. The determined references can be tracked by advanced controllers like MPCs. To obtain an efficient solution while avoiding dramatic changes from the initial solution, a moving horizon approach is developed. A reduced online problem is formulated based on the initial solution.

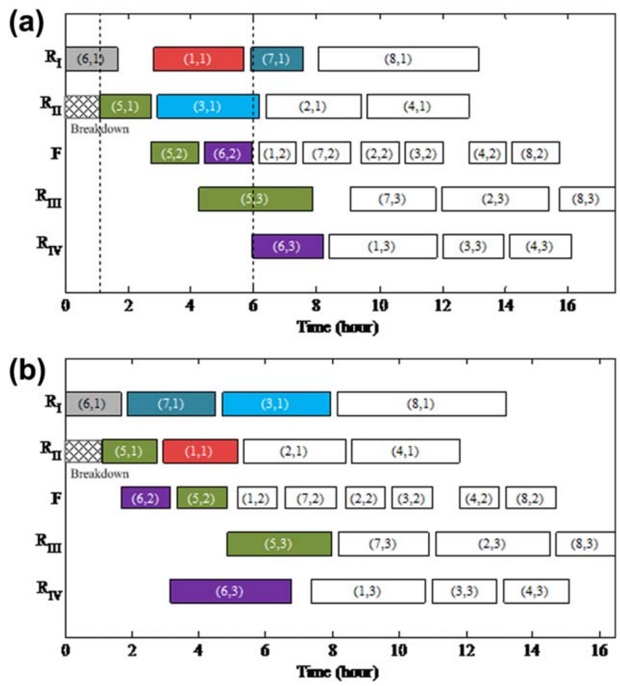


Figure 16. The scheduling results by resolving the integrated problem under uncertainty scenario 2.

(a) Shifted schedule. (b) Online solution. The label on a task bar denotes (job, stage). Reactor R_{II} breaks down from 0 to 1 h. The integrated problem is solved when the reactor is recovered. The period between two dash lines in the shifted schedule is the resolving horizon. The historical tasks are represented in gray and the future tasks beyond the resolving horizon are represented by white boxes. The tasks in the resolving horizon are colored. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

The reduced problem is much simpler than the entire integrated problem and can be solved efficiently. As only a part of the initial solution is changed, rescheduling stability is guaranteed. The online integrated method is demonstrated by the case study including eight jobs and 32 dynamic models. The online integrated problem in a resolving horizon of 1 h can be solved efficiently in 3 min, whereas solving the entire integrated problem requires more than 13 h. Under uncertainties of the control system disruptions and the processing unit breakdown, the online solution prevents a large loss in the production profit. In the worst case, the decreased profit is only 1.5% for the integrated method, whereas it is 12.1% for the shifted rescheduling method. In this work, we do not consider the sequence-dependent changeovers, which will be formulated in the integrated problem for the future research.

Table 2. Comparisons of the Shifted Schedule and the Online Solution of the Integrated Problem Under Uncertainty Scenario 2

	Shifted Schedule	Online Solution
Profit (m.u.)	377.6	422.7
Sales (m.u.)	720.8	769.7
Processing cost (m.u.)	103.2	107.0
Fixed cost (m.u.)	240	240

Notation

Set

$\underline{I_{jk}}$ = units capable of processing task (j,k)
$\underline{JK}$ = tasks in historical period
$\overline{JK}$ = tasks in future period beyond resolving horizon
JK = tasks in resolving horizon, $JK = JK^{F} \cup JK^{D}$
JK^{D} = tasks with dynamic models in resolving horizon
JK^{F} = tasks with fixed recipe in resolving horizon
$\overline{JK}^{D}$ = tasks with dynamic models in future period
$\overline{JK}^{F}$ = tasks with fixed recipe in future period
K_j = operational stages for job j

Index

i = processing unit
j, j' = job
k, k' = stage
$(j,k), (j',k')$ = task
r = finite element
q, q' = collocation point

Binary variable

ξ_{ijk} = equal to one if task (j,k) is assigned to unit i
$\beta_{ijkj'k'}$ = equal to one if task (j,k) precedes task (j',k') in unit i
γ_j = equal to one if job j is completed after d_j

Variable

Cost^{V} = total variable cost
DS_{jk} = absolute change in starting time of task (j,k)
DT_j = completion time of job j
DT_j^{I}, DT_j^{II} = components of DT_j
L_{ijk} = length of finite element in dynamic model for task (j,k) in unit i
PC_{ijk} = variable processing cost of task (j,k) executed in unit i
PR_j = price value of job j
Profit = production profit
PT_{ijk} = variable processing time of task (j,k) executed in unit i
Sales = sales of jobs
T_{ijk} = time variable of dynamic model for task (j,k) in unit i
T_{ijk}^{r} = starting time of finite element r for dynamic model of task (j,k) in unit i
T_{ijk}^{rq} = time point of collocation point q in finite element r for dynamic model of task (j,k) in unit i
TS_{jk} = starting time of task (j,k)
TE_{jk} = ending time of task (j,k)
U_{ijk} = input vector of dynamic model for task (j,k) in unit i
U_{ijk}^{rq} = discretized input vector U_{ijk} at collocation point q in finite element r
VST = total change in task starting times
X_{ijk} = state vector of dynamic model for task (j,k) in unit i
X_{ijk}^{r} = discretized state vector X_{ijk} at beginning of finite element r
X_{ijk}^{rq} = discretized state vector X_{ijk} at collocation point q in finite element r
XPC_{ijk} = product of $\xi_{ijk}\mathrm{PC}_{ijk}$
XPT_{ijk} = product of $\xi_{ijk}\mathrm{PT}_{ijk}$

Parameter

$a_{qq'}$ = collocation matrix
b_m = big-M value
b_q = collocation vector
c_q = collocation vector
c^{F} = total fixed cost
$c_j^{\max}$ = upper bound of completion time for job j
d_j = threshold in price function of job j
l_h = length of resolving horizon
n_{ijk}^{r} = number of finite elements of dynamic model for task (j,k) in unit i
pr_j = constant price for job j
pt_{ijk} = fixed processing time of task (j,k) in unit i
$\mathrm{pt}_{ijk}^{\max}$ = upper bound for processing time of task (j,k) in unit i
s_h = beginning of resolving horizon
x_{ijk}^{I} = initial condition of dynamic model for task (j,k) in unit i
x_{ijk}^{F} = final value of dynamic model for task (j,k) in unit i

Literature Cited

1. Grossmann I. Enterprise-wide optimization: a new frontier in process systems engineering. *AIChE J*. 2005;51:1846–1857.
2. Grossmann IE. Advances in mathematical programming models for enterprise-wide optimization. *Comput Chem Eng*. 2012;47:2–18.
3. Wassick JM, Agarwal A, Akiya N, Ferrio J, Bury S, You FQ. Addressing the operational challenges in the development, manufacture, and supply of advanced materials and performance products. *Comput Chem Eng*. 2012;47:157–169.
4. Yue D, You F. Sustainable scheduling of batch processes under economic and environmental criteria with MINLP models and algorithms. *Comput Chem Eng*. 2013;54:44–59.
5. Engell S, Harjunkoski I. Optimal operation: scheduling, advanced control and their integration. *Comput Chem Eng*. 2012;47:121–133.
6. Harjunkoski I, Nystrom R, Horch A. Integration of scheduling and control—theory or practice? *Comput Chem Eng*. 2009;33:1909–1918.
7. Munoz E, Capon-Garcia E, Moreno-Benito M, Espuna A, Puigjaner L. Scheduling and control decision-making under an integrated information environment. *Comput Chem Eng*. 2011;35:774–786.
8. Chu Y, You F. Integration of scheduling and control with online closed-loop implementation: fast computational strategy and large-scale global optimization algorithm. *Comput Chem Eng*. 2012;47:248–268.
9. Flores-Tlacuahuac A, Grossmann IE. Simultaneous cyclic scheduling and control of a multiproduct CSTR. *Ind Eng Chem Res*. 2006;45:6698–6712.
10. Nystrom RH, Franke R, Harjunkoski I, Kroll A. Production campaign planning including grade transition sequencing and dynamic optimization. *Comput Chem Eng*. 2005;29:2163–2179.
11. Prata A, Oldenburg J, Kroll A, Marquardt W. Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. *Comput Chem Eng*. 2008;32:463–476.
12. Zhuge J, Ierapetritou MG. Integration of scheduling and control with closed loop implementation. *Ind Eng Chem Res*. 2012;51:8550–8565.
13. Mishra BV, Mayer E, Raisch J, Kienle A. Short-term scheduling of batch processes. A comparative study of different approaches. *Ind Eng Chem Res*. 2005;44:4022–4034.
14. Nie YS, Biegler LT, Wassick JM. Integrated scheduling and dynamic optimization of batch processes using state equipment networks. *AIChE J*. 2012;58:3416–3432.
15. Chu Y, You F. Integrated scheduling and dynamic optimization of complex batch processes with general network structure using a generalized Benders decomposition. *Ind Eng Chem Res*. 2013;52:7867–7885.
16. Angel Gutierrez-Limon M, Flores-Tlacuahuac A, Grossmann IE. A multiobjective optimization approach for the simultaneous single line scheduling and control of CSTRs. *Ind Eng Chem Res*. 2012;51:5881–5890.
17. Flores-Tlacuahuac A, Grossmann IE. Simultaneous cyclic scheduling and control of tubular reactors: single production lines. *Ind Eng Chem Res*. 2010;49:11453–11463.
18. Capon-Garcia E, Moreno-Benito M, Espuna A. Improved short-term batch scheduling flexibility using variable recipes. *Ind Eng Chem Res*. 2011;50:4983–4992.
19. Chu Y, You F. Integration of production scheduling and dynamic optimization for multi-product CSTRs: generalized benders decomposition coupled with global mixed-integer fractional programming. *Comput Chem Eng*. 2013;58:315–333.
20. Chu Y, You F. Integration of scheduling and dynamic optimization of batch processes under uncertainty: two-stage stochastic programming approach and enhanced generalized benders decomposition algorithm. *Ind Eng Chem Res*. 2013;52:16851–16869.
21. Barton PI, Allgor RJ, Feehery WF, Galan S. Dynamic optimization in a discontinuous world. *Ind Eng Chem Res*. 1998;37:966–981.
22. Chachuat B, Singer AB, Barton PI. Global mixed-integer dynamic optimization. *AIChE J*. 2005;51:2235–2253.
23. Sand G, Engell S. Modeling and solving real-time scheduling problems by stochastic integer programming. *Comput Chem Eng*. 2004;28:1087–1103.
24. Sahinidis NV. Optimization under uncertainty: state-of-the-art and opportunities. *Comput Chem Eng*. 2004;28:971–983.
25. Verderame PM, Elia JA, Li J, Floudas CA. Planning and scheduling under uncertainty: a review across multiple sectors. *Ind Eng Chem Res*. 2010;49:3993–4017.
26. Chu YF, Wassick JM, You FQ. Efficient scheduling method of complex batch processes with general network structure via agent-based modeling. *AIChE J*. 2013;59:2884–2906.
27. Chu Y, You F, Wassick JM. Hybrid method integrating agent-based modeling and heuristic tree search for scheduling of complex batch processes. *Comput Chem Eng*. 2014;60:277–296.
28. Yue D, You F. Planning and scheduling of flexible process networks under uncertainty with stochastic inventory: MINLP models and algorithm. *AIChE J*. 2013;59:1511–1532.
29. Wu SD, Storer RH, Chang PC. One-machine rescheduling heuristics with efficiency and stability as criteria. *Comput Oper Res*. 1993;20:1–14.
30. Aytug H, Lawley MA, McKay K, Mohan S, Uzsoy R. Executing production schedules in the face of uncertainties: a review and some future directions. *Eur J Oper Res*. 2005;161:86–110.
31. Kopanos GM, Capon-Garcia E, Espuna A, Puigjaner L. Costs for rescheduling actions: a critical issue for reducing the gap between scheduling theory and practice. *Ind Eng Chem Res*. 2008;47:8785–8795.
32. Mendez CA, Cerda J. An MILP framework for batch reactive scheduling with limited discrete resources. *Comput Chem Eng*. 2004;28:1059–1068.
33. Dimitriadis AD, Shah N, Pantelides CC. RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. *Comput Chem Eng*. 1997;21:S1061–S1066.
34. Beraldi P, Ghiani G, Grieco A, Guerriero E. Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. *Comput Oper Res*. 2008;35:3644–3656.
35. Harjunkoski I, Grossmann IE. Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. *Comput Chem Eng*. 2002;26:1533–1552.
36. Mendez CA, Cerda J. Dynamic scheduling in multiproduct batch plants. *Comput Chem Eng*. 2003;27:1247–1259.
37. Demir Y, Isleyen SK. Evaluation of mathematical models for flexible job-shop scheduling problems. *Appl Math Model*. 2013;37:977–988.
38. Hairer GWE. *Solving Ordinary Differential Equations II*. Springer, Berlin, Germany, 2010.
39. Chu Y, You F. Integrated scheduling and dynamic optimization of sequential batch processes with online implementation. *AIChE J*. 2013;59:2379–2406.
40. Rosenthal RE. *GAMS—A User's Guide*. Washington, DC: GAMS Development Corporation, 2013.
41. Luyben WL. *Chemical Reactor Design and Control*. Wiley-Interscience, Washington, DC, USA, 2007.

Appendix Parameters for Case Studies

Table A1. Parameters of the Scheduling Model

	Job 1, 2, 3, 4	Job 5, 6, 7, 8
pr_j (m.u.)	100	100
d_j (h)	15	15
c_j^{max} (h)	20	20
pt_{ijk} (filtration tasks) (h)	1.2	1.5
$cost_j^{fix}$ (m.u.)	30	30

The task changeover time in each processing unit is 0.2 h. The processing units are ready at the beginning so that there is no changeover time for the first task executed in a unit.

Table A2. Parameters of Dynamic Models for Reaction Tasks

Job	Job 1, 2, 3, 4		Job 5, 6, 7, 8	
Stage	Stage 1	Stage 3	Stage 1	Stage3
k_S (h^{-1})	2E1	1E2	1E4	5E2
e_S (K)	1E3	1.5E3	3E3	2E3
c_S^0 (kmol/m^3)	1	0.9	1	0.9
c^R (m.u.)	0.2	0.2	0.2	0.2
t_R^{min} (K)	300	300	300	300
t_R^{max} (K)	350	350	350	350
χ_S (%)	90	90	90	90
t_R^{final} (K)	300	300	300	300
# Finite elements	30	30	30	30

Table A3. Parameters of the Temperature Control System

Symbol	Description	Value	Unit
s_{TC}	Control horizon	1	H
v_R	Volume of reactor	3	m^3
ρ_R	Density of reactor	8E2	kg/m^3
cp_R	Heat capacity of reactor	3.5	kJ/(kg K)
ua_J	Heat transfer coefficient	1E5	kJ/(h K)
v_J	Volume of jacket	1	m^3
ρ_J	Density of jacket	1E3	kg/m^3
cp_J	Heat capacity of jacket	4.2	kJ/(kg K)
t_J^H	Heating fluid temperature	370	K
t_J^C	Cooling fluid temperature	280	K
t_R^{min}	Minimum reactor temperature	300	K
t_R^{max}	Maximum reactor temperature	370	K
f_{JH}^{min}	Minimum heating flow rate	0	m^3/h
f_{JH}^{max}	Maximum heating flow rate	20	m^3/h
f_{JC}^{min}	Minimum cooling flow rate	0	m^3/h
f_{JC}^{max}	Maximum cooling flow rate	20	m^3/h

Manuscript received Jun. 18, 2013, and revision received Dec. 23, 2013.