A Novel Adaptive Surrogate Modeling-Based Algorithm for Simultaneous Optimization of Sequential Batch Process Scheduling and Dynamic Operations

Hanyu Shi and Fengqi You
Dept. of Chemical and Biological Engineering, Northwestern University, Evanston, IL 60208

DOI 10.1002/aic.14974
Published online August 12, 2015 in Wiley Online Library (wileyonlinelibrary.com)

A novel adaptive surrogate modeling-based algorithm is proposed to solve the integrated scheduling and dynamic optimization problem for sequential batch processes. The integrated optimization problem is formulated as a large scale mixed-integer nonlinear programming (MINLP) problem. To overcome the computational challenge of solving the integrated MINLP problem, an efficient solution algorithm based on the bilevel structure of the integrated problem is proposed. Because processing times and costs of each batch are the only linking variables between the scheduling and dynamic optimization problems, surrogate models based on piece-wise linear functions are built for the dynamic optimization problems of each batch. These surrogate models are then updated adaptively, either by adding a new sampling point based on the solution of the previous iteration, or by doubling the upper bound of total processing time for the current surrogate model. The performance of the proposed method is demonstrated through the optimization of a multi-product sequential batch process with seven units and up to five tasks. The results show that the proposed algorithm leads to a 31% higher profit than the sequential method. The proposed method also outperforms the full space simultaneous method by reducing the computational time by more than four orders of magnitude and returning a 9.59% higher profit. © 2015 American Institute of Chemical Engineers *AIChE J*, 61: 4191–4209, 2015

Keywords: adaptive surrogate modeling, scheduling, dynamic optimization, integrated optimization, sequential batch processes

Introduction

Increasingly fierce competition between the modern enterprises is driving the process industries to optimize their manufacturing processes to respond to eroding profit margins.[1–3] To remain competitive in the global marketplace, process companies have attached more importance to integrated optimization of decision making across multiple levels.[4–6] Thus, reliable and efficient optimization strategies and solution methods for integrated optimization problems have become essential.[7–14] In this work, we address the integrated optimization of scheduling and dynamic optimization problems for a sequential batch process, in which the integrity of each batch unit is preserved. The objective is to maximize the total profit over the entire production horizon.[15–18]

Previous work has been reported in the literature on the integrated batch scheduling problem for complex batch distillation and batch reactor/distillation processes.[19,20] In addition, mathematical programming approach and efficient solution methodology have been proposed to determine the design parameters and operating policy for batch distillation processes simultaneously.[21,22] In this work, we consider a more general issue, the integrated optimization of scheduling and dynamic optimization problems for a sequential batch process. A monolithic model for the integrated scheduling and dynamic optimization of a sequential batch process can be formulated by combining a scheduling model and a set of dynamic optimization problems.[23–25] In general, a scheduling problem is solved to assign limited resources to manufacture a set of products in a specific time horizon,[26,27] and the batch manufacturing process needs to follow certain batch recipes.[28,29] The dynamic optimization problem is solved to optimize the operating trajectories of dynamic processes, for example, chemical reactions or transition processes. As the production scheduling models are usually mixed-integer optimization problems, the integrated optimization problem is formulated as a mixed-integer dynamic optimization (MIDO) problem.[30,31] The MIDO problem can then be reformulated into a mixed-integer nonlinear programming (MINLP) problem using the orthogonal collocation method.[32,33] However, the resulting MINLP problem is usually large in scale and computationally expensive to solve with general purpose MINLP solvers.[34–36] The aim of this work is to address this computational challenge and to propose an efficient solution method to solve this integrated scheduling and dynamic optimization problem for the sequential batch process.

Flexible recipe methods have been widely used to the batch scheduling problem to obtain enhance the flexibility of the resulting production schedule.[37–40] By employing the powerful surrogate modeling approach for representing complex

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

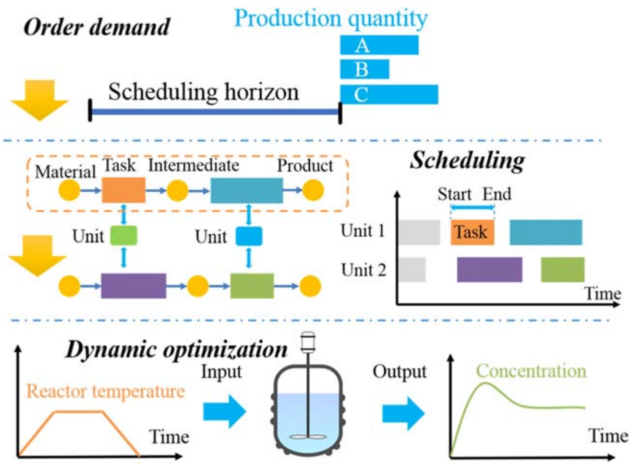


Figure 1. Scope of integrated problem.

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

systems and thereby reducing computational complexity,[41–48] we propose a novel adaptive surrogate modeling-based algorithm to solve the integrated scheduling and dynamic optimization problem for sequential batch processes. In this integrated optimization problem for sequential batch processes, the scheduling problem is linked with dynamic optimization problems only via operating recipes, which include batch processing times and processing costs. We first exploit the special bilevel structure of the integrated optimization problem and use a set of piecewise linear surrogate models to replace the nonlinear dynamic optimization problems in the integrated optimization problem. To update the surrogate models adaptively, we either add a new sampling point based on the solution of the previous iteration or double the upper bound of total processing time for the current surrogate model. The algorithm terminates until all the solution points satisfy two stopping criterions. To satisfy the two stopping criterions, the solution point should be located within a certain distance of one existing sampling point but not located exactly on the last sampling point. The performance of the adaptive surrogate modeling-based algorithm is demonstrated through a case study of a sequential batch process with seven units for three production lines and up to five operational tasks for each production line. The results of the proposed method are compared with those obtained from the sequential method and the simultaneous method.

The main novelties of this work are summarized as follows:

- Novel piecewise linear surrogate models replacing the nonlinear dynamic models in the integrated scheduling and dynamic optimization problem with much higher computational efficiency.
- A novel approach to adaptively update the surrogate model enabling the surrogate models maintain high fidelity with only a small set of sampling points.

The remainder of this work is structured as follows. The problem statement is given in the next section. The model formulation of the integrated scheduling and dynamic optimization problem is presented next as well as the special bilevel structure of the integrated problem. In the Adaptive Surrogate Modeling-Based Algorithm section, we show a general three-step procedure to build a piecewise linear surrogate model and the adaptive algorithm to update the surrogate models. In the Case Study section, we use a case study of one multiproduct sequential batch process to demonstrate the performance of the proposed adaptive surrogate modeling based algorithm. Finally, we present our conclusions and give directions for future research.

Problem Statement

In this work, we focus on sequential batch processes where the integrity of each batch unit is preserved within the production process.[26,27] Namely, operations of batch splitting, mixing, and resizing are prohibited in the sequential batch process.

As shown in Figure 1, there are order demands with specific due dates for a set of products. Because the integrity of the batch is preserved in the sequential batch process, the number of the batches for one product is integral. There are several operational stages within a batch process according to different routing pathways. The production time of each unit is divided into several time slots. The scheduling problem is solved to determine the starting time of each stage and the assignment of a capable unit to each stage in a batch.[49,50] The dynamic optimization problem is solved to obtain the optimal processing time and processing cost of each operational stage. Following the approach widely used in industry, we introduce price penalties for a tardy orders.[51] Specifically, the demand should be fulfilled before its due date; otherwise the price will decrease linearly with respect to time. As shown in Figure 2, there are two due dates for each batch. The batch price stays constant if the batch is delivered before the first due date. The order price decreases with time linearly if it is delivered between the first and second due dates, and is zero if it is delivered after the second due date.

In this problem, we consider a sequential batch process, where operations of batch splitting, mixing, and resizing are not allowed.[26,27] We are given the following parameters:

- Production configuration, including the routing pathway for each batch;
- Upper bound of the scheduling horizon;
- Divided batches with due dates;
- Unit costs of raw materials and utilities;
- Dynamic models of optional stages;
- Initial conditions in the first stage and final values in the last stage for each batch;

Utility constraints, safety constraints, and quality constraints.

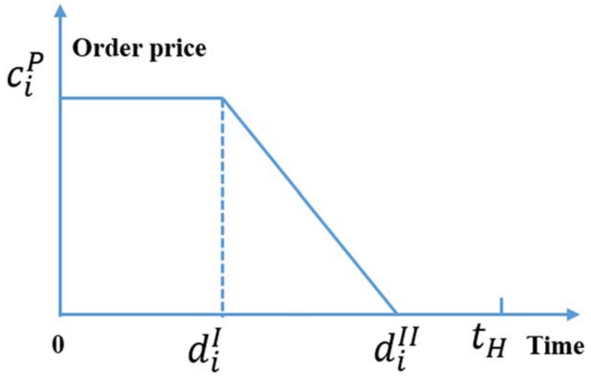


Figure 2. The batch price function.

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

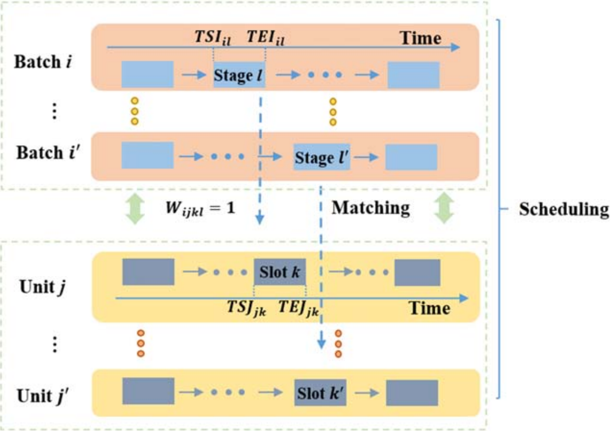


Figure 3. Time slot model.

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

Major decision variables for these problems are:

- Starting time of each operational stage of each batch;
- Assignment of a processing unit to each stage of each batch;
- Delivery date and economic value of each batch;
- Processing time and processing cost of each operational stage.

The objective of this problem is to maximize the total production profit over the scheduling horizon.

Model Formulation

The integrated problem for scheduling and dynamic optimization of the sequential batch process is presented in this section. For batch processes, the scheduling problem is solved to allocate limited raw materials and batch units to manufacture a set of products which may follow different production recipes.[26,27] Dynamic optimization problems are solved to determine the optimal dynamic operating profiles for chemical reaction operations.[52] In the conventional sequential solution approach, the dynamic optimization problems are solved first to obtain a set of fixed operational recipes, based on which the scheduling problem is then optimized.[16] However, fixed recipes are not able to capture the dynamics of the batch processes. The integrated optimization of batch production scheduling and dynamic operations is essential to optimize the overall performance of batch process.

Model formulation for production scheduling of sequential batch processes

An illustration of the scheduling problem is shown in Figure 3, which is based on the concept of time slots.[26,27] A set of stages indexed by l are included in each batch indexed by i. A set of time slots indexed by k is contained in capable units indexed by j. With the time-slot formulation, the main scheduling decision is to assign a time slot of a capable unit to produce one stage of a batch. The binary variable W_{ijkl} is introduced to determine this assignment. If stage l of batch i is produced in slot k of unit j, W_{ijkl} is equal to 1 and the starting time TSI_{il} of stage l in batch i is equal to the starting time TSJ_{jk} of slot k in unit j.

The detailed scheduling model can be found in Appendix A.[23,53] Here, we only present the main ideas of the scheduling model for the sequential batch process. The entire scheduling model includes four sections: unit allocation section, timing matching section, batch price function section, and sales and costs section. In the unit allocation section, a binary variable is introduced to determine the assignment of all stages to capable units. The timing matching section is used to determine the starting time and the end time of each slot in one unit and the duration of each stage of each batch. In the batch price function, a penalty is charged by the tardy batch and the selling price decreases with respect to time. Sales can be obtained by summing up all the batch prices. The total cost is taken as the sum of the individual batch cost.

Model formulation for dynamic optimization

The detailed model formulation of the dynamic optimization problem is given in Appendix B.[23] The dynamic optimization model contains a set of differential equations. We adopt the collocation method to discretize the differential and algebraic equations.[54] The collocation approach is an implicit Runge–Kutta method.[55] For dynamic stage l of batch i, we first use a set of equal-length finite elements to partition the time horizon into a number of time intervals. We note that the lengths of these time intervals are variables, and in the same stage they are the same to each other. In each finite element, several collocation points are chosen according to the specific collocation method.[56] We then model the time-dependent variables, including the state variables and the input variables, at those discrete points. With this discretization process, the differential equations are reformulated as a set of algebraic equations. The discretization process is demonstrated as follows.

In Eq. 1, the time horizon for stage l of batch i is divided by a set of grid points, T_{il}^{rq}, where r is the index for each finite element, q represents the index for the collocation points on each finite element, and Δ_{il} denotes the length of the finite element in dynamic stage l of batch i. Constraint (2) represent the time constraints for the discretized time variables

$$T_{il}^{rq} = T_{il}^{r-1} + c_q \Delta_{il}, \quad \forall i, l \in L_i, r > 0, q \tag{1}$$

$$0 \le T_{il}^{rq} \le \tau_{il}, \quad \forall i, l \in L_i, r, q \tag{2}$$

In Eqs. 3 and 4, the state variables are discretized and approximated by the collocation method on the corresponding grid points. The coefficients, including $a_{qq'}$, c_q, and b_q, are provided by the Runge–Kutta method

$$X_{il}^{rq} = X_{il}^{r-1} + \Delta_{il} \sum_{q=1}^{n_q} a_{qq'} f_{il}\left(X_{il}^{rq'}, U_{il}^{rq'}\right), \quad \forall i, l \in L_i, r > 0, q \tag{3}$$

$$X_{il}^{r} = X_{il}^{r-1} + \Delta_{il} \sum_{q=1}^{n_q} b_q f_{il}\left(X_{il}^{rq}, U_{il}^{rq}\right), \quad \forall i, l \in L_i, r > 0 \tag{4}$$

Alongside the differential equations, the path constraints, the output variables, and the criterion equations are also transformed into discretized values, as shown in Eqs. 5–7

$$h_{il}\left(X_{il}^{rq}, U_{il}^{rq}\right) \le 0, \quad \forall i, l \in L_i, r > 0, q \tag{5}$$

$$Y_{il}^{rq} = g_{il}\left(X_{il}^{rq}, U_{il}^{rq}\right), \quad \forall i, l \in L_i, r > 0, q \tag{6}$$

$$\varphi_{il} = \phi_{il}\left(Y_{il}^{rq}\right), \quad \forall i, l \in L_i \tag{7}$$

Equation 8 defines the initial values of state variables for each dynamic stage of batch i. The initial values for the first dynamic stage are set as predetermined parameters. However, the initial values for other dynamic stages are set to be the same as the final values of the dynamic model in the previous stage, that is, the state values at the first grid point in stage l are equivalent to the state values at the last collocation point of the last finite element in stage $l-1$. n_r is the total number of finite elements in one dynamic stage. n_q is the total number of collocation points in one finite element

$$X_{il}^{(r=0)}=\begin{cases} X_{i(l-1)}^{(r=n_r)(q=n_q)}, & l\geq 2 \\ x_i^0, & l=1 \end{cases}, \quad \forall i,l\in L_i \tag{8}$$

Equation 9 defines that the final value of the state variables for batch i is the same as the value in the last collocation point of the last finite element in last dynamic stage for batch i. The new discretized quality constraint is shown in constraint (10)

$$X_i^f=X_{il}^{(r=n_r)(q=n_q)}, \quad l=|L_i|, \forall i \tag{9}$$

$$q_i\left(X_{il}^{(r=n_r)(q=n_q)}\right)\leq 0, \quad l=|L_i|, \forall i \tag{10}$$

Model formulation and properties of the integrated optimization problem

According to the aforementioned model formulations,[23,53] the production scheduling problem needs the operating recipes, including the processing times and processing costs, for each batch stage to determine the scheduling sequence over the scheduling horizon. In the conventional hierarchical method, the operating recipes in the scheduling problem are often set as fixed parameters, obtained by solving the dynamic optimization problems in advance. However, the fixed operating recipes might not be optimal for the scheduling problem.[7] In this work, this problem is solved with an integrated method, in which the process dynamic operating profiles are optimized simultaneously with the production scheduling decisions. The integrated method links the scheduling model and dynamic model together by linking Eqs. 11 and 12, where PT_{il} and PC_{il} are the processing time and processing cost for stage l of batch i in scheduling model, and τ_{il} and φ_{il} are the corresponding processing time and processing cost obtained from the dynamic optimization problem

$$PT_{il}=\tau_{il}, \quad \forall i,l\in L_i \tag{11}$$

$$PC_{il}=\varphi_{il}, \quad \forall i,l\in L_i \tag{12}$$

The full space integrated scheduling and dynamic optimization problem is formulated as follows:

(Integrated_MIDO)
max *Profit* in Eq. A20
s.t. Scheduling model constraints (A1)–(A19)
Dynamic model constraints (B1)–(B9)
Linking equations 11 and 12

This is an MIDO problem, which is usually reformulated into an MINLP problem based on the collocation method.[33] The integrated MINLP problem for the sequential batch process is presented as follows:

(Integrated_MINLP)
max *Profit* in Eq. A20
s.t. Scheduling model constraints (A1)–(A19)
Dynamic model constraints (1)–(10)
Linking equations 11 and 12

However, the integrated MINLP problem is usually large in scale and difficult to solve with general purpose MINLP solvers. A special property of the **(Integrated_MINLP)** problem is that the scheduling problem is linked with the dynamic optimization problems only via two simple linking equations. To have a clear view of the structure for the integrated optimization problem, we first divide the decision variables into four categories: variables in the scheduling level, V^{Sch}; variables only in the dynamic model of batch i, V_i^{Dyn}; the collection of processing times $\{PT_{il}\}$; and the collection of processing cost $\{PC_{il}\}$. We can then rewrite the objective of the integrated problem as follows

$$Profit^*=\max_{\substack{\{PT_{il}\},\forall i,l\in L_i;\\ \{PC_{il}\},\forall i,l\in L_i;\\ V^{\mathrm{Sch}};\\ \{V_i^{\mathrm{Dyn}}\},\forall i}} Profit \tag{13}$$

According to Eq. 13, we can separate the four variable categories into two types: the variables that can be placed in the outer level problem and the variables that can be placed in the inner level problem, as shown in Eq. 14

$$Profit^*=\max_{\substack{\{PT_{il}\},\forall i,l\in L_i;\\ V^{\mathrm{Sch}}}}\left\{\max_{\substack{\{PC_{il}\},\forall i,l\in L_i;\\ \{V_i^{\mathrm{Dyn}}\},\forall i}}\{Sales-Cost\}\right\} \tag{14}$$

In Eq. 14, the processing cost variables and the decision variables for each batch in the dynamic optimization problems are in the inner level problems, while the processing time variables and the decision variables for the scheduling level problem are in the outer level problem. In the inner level problem, the decision variables from the outer level problem, including the processing time variables and other scheduling variables, are regarded as fixed parameters. According to Eqs. A16 and A18, we can see that the *Sales* and the fixed cost are independent of the processing cost variables and the dynamic optimization variables. Thus, they can be moved to the outer level problem. In the inner level problem, the scheduling variables are regarded as fixed parameters, so the dynamic model of one batch is independent from those of another batch. This model property allows us to reformulate the objective of the bilevel integrated problem as Eq. 15

$$Profit^*=\max_{\substack{\{PT_{il}\},\forall i,l\in L_i;\\ V^{\mathrm{Sch}}}}\left\{Sales-c^{\mathrm{Fix}}-\sum_i\min_{\substack{\{PC_{il}\},\forall i,l\in L_i;\\ \{V_i^{\mathrm{Dyn}}\},\forall i}}\left\{\sum_{l\in L_i}PC_{il}\right\}\right\} \tag{15}$$

Equation 15 indicates that the dynamic optimization problem for each batch can be solved independently. This independence also results from the network structure of the sequential batch process, where operations of batch splitting, mixing, and resizing are prohibited. We rewrite the integrated scheduling and dynamic optimization problem for sequential batch processes into the following bilevel programming problem.

(Bilevel_Integrated)

$$\text{(Outer)}\ \max_{\substack{\{PT_{il}\},\forall i,l\in L_i;\\ V^{\mathrm{Sch}}}}\left\{Sales-c^{\mathrm{Fix}}-\sum\nolimits_i P_i^c\right\} \tag{16}$$

s.t. Scheduling model equations 5–A19)

$$P_i^C = \eta_i\left(PT_{i1}, PT_{i2}, \ldots PT_{i|L_i|}\right), \quad \forall i, l \in L_i \tag{17}$$

(Inner) Inner level optimization for each batch i

$$\eta_i\left(PT_{i1}, PT_{i2}, \ldots PT_{i|L_i|}\right) = \min_{V_i^{\mathrm{Dyn}}} \quad \sum\nolimits_{l \in L_i} PC_{il} \tag{18}$$

s.t. Dynamic model (1)–(10)
Linking equations 11 and 12

In the **(Bilevel_Integrated)** problem, the integrated scheduling and dynamic optimization problem is reformulated into an outer level scheduling problem with the optimal-value function (17) and the inner level dynamic optimization problem. In the inner level problems, the processing times are treated as fixed parameters. The inner level problems are a set of multistage dynamic optimization problems. The objective of inner level problems is to minimize the total processing cost over all stages at the given processing time. The optimal objective value in the inner level problem is a function of processing time, which is referred to as the optimal-value function (18) and also appears in the outer level problem as Eq. 17. The optimal-value function should capture the details of the corresponding dynamic optimization problem. The outer level problem is a scheduling problem with optimal-value functions. In the outer level problem, the scheduling problem is solved with the operating recipes that are provided by the optimal-value functions. Based on the bilevel structure of the integrated problem, we propose an adaptive surrogate-based algorithm to enhance the computational efficiency.

Adaptive Surrogate Modeling-Based Algorithm

The integrated scheduling and dynamic optimization problem for the sequential batch process is a large scale MINLP problem that is difficult to solve by general purpose MINLP solvers.[57] However, the integrated optimization problem has a special bilevel structure where the scheduling problem and the dynamic optimization problems are connected only by operating recipes, which include both processing costs and times. Based on the structure of the integrated problem, we replace the dynamic optimization problems with a set of piecewise linear surrogate models, which could capture the main characteristics of the dynamic optimization problems.[16,58,59] To maintain fidelity of those surrogate models at the solution points, we propose a novel strategy to update the surrogate models adaptively.

Surrogate model construction

There are three general steps to construct a surrogate model according to previous literature.[58–64] The first step is to choose an experimental design for generating data. In the **(Bilevel_Integrated)** optimization problem, the inner level dynamic optimization problems only provide the optimal operating recipes, which are processing times and processing costs, to the outer level scheduling problem via the optimal-value function. As a result, the data need to be generated in the experimental design are the operating recipes. The optimal operating recipes for certain total processing time can be determined by the following ε-constraint method.[23] For a total processing time ε_i for batch i, the ε-constraint is shown as the inequality (19)

$$\sum_{l \in L_i} PT_{il} \le \varepsilon_i \tag{19}$$

The processing times and costs for each stage of batch i are obtained in the following **(ε-constraint)** problem[23]

(ε-constraint)

$$PT_{i1}(\varepsilon_i), PT_{i2}(\varepsilon_i), \ldots PT_{i|L_i|}(\varepsilon_i) = \arg \quad \min\nolimits_{\mathrm{PT}_{il}} \sum\nolimits_{l \in L_i} PC_{il}, \\ \forall i P_i^C(\varepsilon_i) = \min_{\mathrm{PT}_{il}} \quad \sum\nolimits_{l \in L_i} PC_{il}, \quad \forall i \tag{20}$$

s.t. Dynamic model (1)–(10)
Linking equations 11 and 12
ε-constraint (19)

The second step of surrogate modeling is to choose a model to represent the data. In general, there are several simple basis functions that we can use to build the surrogate models,[58] such as polynomial functions, exponential and logarithmic function, radial basis functions model,[65] and so forth. In this work, to maintain linearity of the model formulation, we choose piecewise linear function to build the adaptive surrogate model.

The third step to build the surrogate model is fitting the model to the observed data. In this work, piecewise linear functions, modeled by the convex combination method are applied to approximate the recipe function.[66–69] Solving a number of dynamic optimization problems leads to a set of recipe data points $\{\varepsilon_i^{(m)}, PT_{il}^{(m)}, P_i^{C(m)}\}$, where m is the index for sampling points. The approximated function is expressed by constraints (21)–(24). R_i^n is the index set of the discrete points for batch i in the nth iteration of the adaptive surrogate modeling-based algorithm

$$\varepsilon_i = \sum_{m \in R_i^n} \sigma_{im} \cdot \varepsilon_i^{(m)}, \forall i \tag{21}$$

$$PT_{il} = \sum_{m \in R_i^n} \sigma_{im} \cdot PT_{il}^{(m)}, \forall i, l \in L_i \tag{22}$$

$$P_i^C = \sum_{m \in R_i^n} \sigma_{im} \cdot P_i^{C(m)}, \forall i \tag{23}$$

$$\sum_{m \in R_i^n} \sigma_{im} = 1, \forall i \tag{24}$$

Constraints (25)–(30) define that $\{\sigma_{i1}, \sigma_{i2}, \ldots, \sigma_{i|R_i^n|}\}$ for any batch i are specially ordered set of Type 2 (SOS2) variables, that is, at most two adjacent terms can be nonzero

$$\sigma_{im} \ge 0, \forall i, m \in R_i^n \tag{25}$$

$$\sigma_{i(m=1)} \le \zeta_{i(m=1)}, \forall i, m \in R_i^n \tag{26}$$

$$\sigma_{im} \le \zeta_{i(m-1)} + \zeta_{im}, \forall i, m \in R_i^n, 2 \le m \le |R_i^n| - 1 \tag{27}$$

$$\sigma_{i\left(m=|R_i^n|\right)} \le \zeta_{i\left(m=|R_i^n|-1\right)}, \forall i \tag{28}$$

$$\sum_{m=1}^{|R_i^n|-1} \zeta_{im} = 1, \forall i \tag{29}$$

$$\zeta_{im} \in \{0, 1\}, \forall i, m \in R_i^n \tag{30}$$

If the processing times are variables, the scheduling model will be nonlinear due to the term $W_{ijkl} \cdot PT_{il}$ in constraint (A7). Because W_{ijkl} is a binary variable, this bilinear term can be linearized using the following constraints (31)–(34)[70]

$$TEJ_{jk} = TSJ_{jk} + \sum_{(i,l) \in IL_j} WT_{ijkl}, \forall j, k \in K_j \tag{31}$$

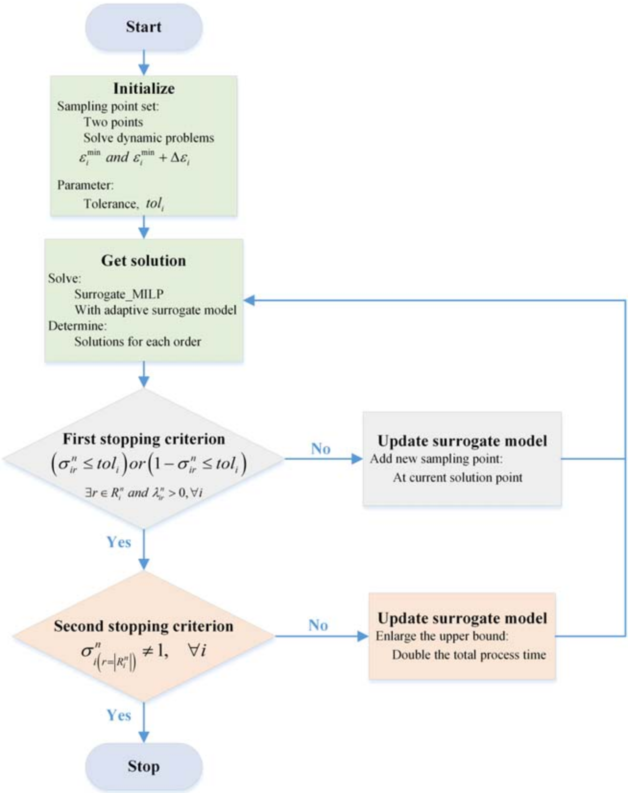

Start
Initialize
Sampling point set:
Two points
Solve dynamic problems
$\varepsilon_i^{\min}$ and $\varepsilon_i^{\min}+\Delta\varepsilon_i$
Parameter:
Tolerance, tol_i
Get solution
Solve:
Surrogate_MILP
With adaptive surrogate model
Determine:
Solutions for each order
First stopping criterion
$(\sigma_{ir}^n \le tol_i) or (1-\sigma_{ir}^n \le tol_i)$
$\exists r \in R_i^n$ and $\lambda_{ir}^n > 0, \forall i$
No
Update surrogate model
Add new sampling point:
At current solution point
Yes
Second stopping criterion
$\sigma_{i(r=|R_i^n|)}^n \ne 1, \quad \forall i$
No
Update surrogate model
Enlarge the upper bound:
Double the total process time
Yes
Stop

Figure 4. Flow chart for adaptive surrogate model.

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

$$WT_{ijkl}+UT_{ijkl}=st_{il}+PT_{il}, \forall i,j \in J_{il}, k \in K_j, l \in L_i \qquad (32)$$

$$0 \le WT_{ijkl} \le W_{ijkl} \cdot t_H, \forall i,j \in J_{il}, k \in K_j, l \in L_i \qquad (33)$$

$$0 \le UT_{ijkl} \le \left(1-W_{ijkl}\right) \cdot t_H, \forall i,j \in J_{il}, k \in K_j, l \in L_i \qquad (34)$$

Thus, the model formulation for integrated scheduling problem with surrogate models is given as follows:

(Surrogate_MILP)

$$\max \quad Sales-c^{\text{fix}}-\sum_i P_i^C \qquad (35)$$

s.t. Scheduling model constraints (A1)–(A6), (A8)–(A19), and (31)–(34)
Sampling points in current iteration
Adaptive surrogate model constraints (21)–(30)

Adaptive surrogate-based optimization algorithm

In this subsection, we discuss the procedure for updating the surrogate model adaptively in order to generate high fidelity surrogate models for the integrated optimization problem. The flow chart for the adaptive surrogate modeling-based algorithm is given in Figure 4. To update the adaptive surrogate model for order i, we start with a simple surrogate model that contains two sampling points. One sampling point is located at the lower bound of the total processing time, determined by solving the dynamic optimization problem directly with the objective to minimize the total processing time, denote as $\varepsilon_i^{\min}$. The location of the second sampling data point is chosen by adding a predetermined time interval $\Delta\varepsilon_i$. The second sampling data point acts as the upper bound of the total processing time for the simple surrogate model. However, the upper bound of the total processing time for the adaptive surrogate model may change because we might need to increase the upper bound of the total processing time in later iterations. This issue will be discussed in a subsequent subsection.

After obtaining the simple surrogate model, we solve the (**Surrogate_MILP**) for the first iteration. We then analyze the solution for the current iteration. The solution point should be located within the solution region defined by the surrogate model. We then check two stopping criterions to determine whether the algorithm should continue for the next iteration. If there is one batch that violates one of the stopping criterions, a new surrogate model should be constructed for this batch and the algorithm should continue. We note that there might be several surrogate models for different batches that need to be updated per iteration.

The first stopping criterion is that the solution point is close enough to one of the existing point. tol_i denotes the predetermined tolerance for the distance between the current solution point and the nearest existing sampling point. If the distance is small enough, it means that the current solution point is located very close to one of the existing sample points. We regard the surrogate model is of fidelity for the current solution point. Otherwise, if the distance is larger than the predetermined tolerance, we will add the new sampling point at the location of this solution point, and build a new adaptive surrogate model.

The first stopping criterion and its corresponding approach to update the surrogate model are illustrated in Figure 5. In Figure 5a, point E is the solution point returned by the current iteration. Point E is located on the piecewise surrogate model that is constructed based on the previous sampling point set. Points B and C are directly next to point E. The length b denotes the processing time interval between point B and E, $b=T_{\text{E}}-T_{\text{B}}$. The length c denotes the processing time interval between point C and E, $c=T_{\text{C}}-T_{\text{E}}$. The length a denotes the processing time interval between point B and C, $a=T_{\text{C}}-T_{\text{B}}$. The first stopping criterion is satisfied only if point E is close enough to one of the points in its immediate vicinity. For batch i, the first stopping criterion can then be expressed as $(b/a \le tol_i,$ or $c/a \le tol_i)$. According to the piecewise linear surrogate model formulation, the first stopping criterion can be expressed as $\left(\sigma_{im}^n \le tol_i\right)$ or $\left(1-\sigma_{im}^n \le tol_i\right), \exists m \in R_i^n$ and $\sigma_{im}^n > 0$, $\forall i$.

If the first stopping criterion is violated, the surrogate model should be updated accordingly, as shown in Figure 5b. If point E is neither close enough to point B nor close enough to point C, the current surrogate model might not be an adequate representation of the corresponding dynamic optimization problem in the vicinity of the current solution point. To overcome this problem, we add a new sampling point at the point with processing time T_{E}. In Figure 5b, the new sampling point is denoted as point F, which has the same processing time as the current solution point E. The new piecewise linear surrogate model is built by adding point F for the next iteration. This approach of adding a new sampling point to refine the model is similar to the branch-and-refine algorithm for global optimization of separable concave minimization problems,[71–74] although it is used in the context of surrogate-based modeling and optimization in this work.

The second stopping criterion is that the solution point is not located at the last sampling point of the current surrogate model. In other words, the total processing time of the solution should not reach the upper bound of the current adaptive

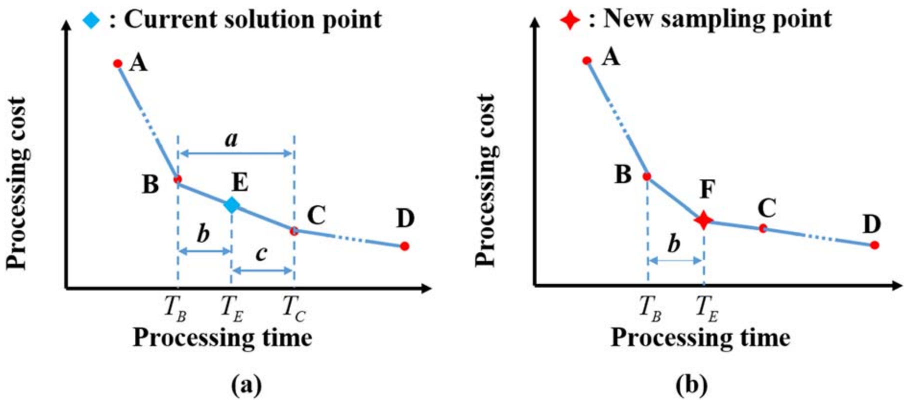


Figure 5. First stopping criterion (a) current solution point violates the first stopping criterion; (b) new sampling point is added to the adaptive surrogate model.

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

surrogate model. If the solution point is at the location of the last sampling point, it is likely that the current upper bound for the total processing time is not large enough. In this case, one more sampling point, which is the new upper bound, is added to build a new adaptive surrogate model. According to the piecewise linear surrogate model formulation, the second stopping criterion can be expressed as $\sigma^n_{i(m=|R_i^n|)} \neq 1, \quad \forall i$.

The second stopping criterion is illustrated in Figure 6a. Point C is the current solution point returned by the **(Surrogate_MILP)** in the latest iteration. The second stopping criterion is that point C is located exactly at the position of the last sampling point in the current sampling point set. If the solution point reaches the upper bound of the surrogate model, the solution point might have the trend to move further to a larger processing time. We note that the lower bound of the surrogate model is fixed, as it is determined by solving the dynamic optimization problem with the objective to minimize the total processing time. To overcome the limitation on the processing time for the solution point, the upper bound of the total processing time in the surrogate model is doubled if the current solution point is located at the last sampling point. In Figure 6b, the new sampling point D is added to the new surrogate model. The processing time of point D is twice as that of point C, $T_D = 2T_C$.

The algorithm terminates when both stopping criterions are satisfied; otherwise, we need to build new adaptive surrogate models and solve the **(Surrogate_MILP)** with new surrogate models for the next iteration. The pseudocode for the adaptive surrogate-based algorithm is given in Figure 7.

Case Study

In this section, we first describe the resource task network representation[75] for the scheduling problem of a sequential batch process and show the differential and algebraic equations for the multistage dynamic optimization problems. To illustrate the details of the experimental design process for generating data of the surrogate model, we take one multistage dynamic optimization problem for the first product as an example. Finally, the performance of the proposed adaptive surrogate modeling-based algorithm is demonstrated. In parallel with our novel solution approach, we also apply the conventional sequential solution method and the full space simultaneous method to this optimization problem for the sequential batch process. The sequential method solves the scheduling problem and the dynamic optimization problems separately, while the simultaneous method formulates the integrated scheduling and dynamic optimization problem into a holistic model and solves the full space MINLP problem directly.[76] The data for the case study can be found in Appendix C.

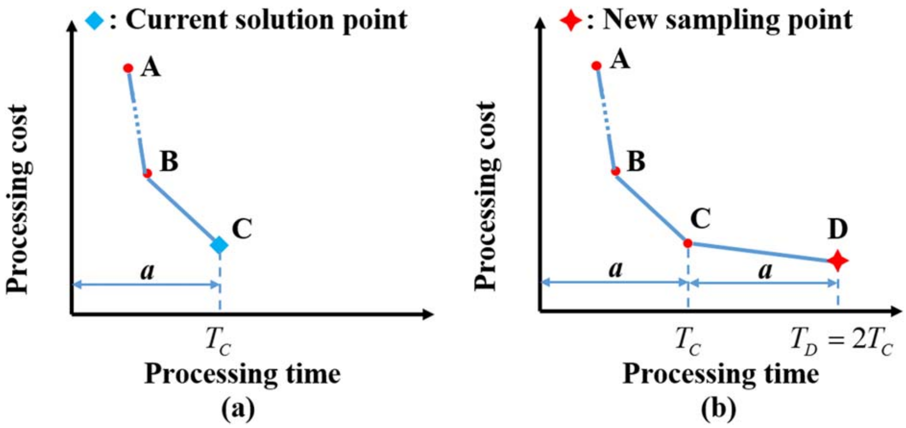


Figure 6. Second stopping criterion (a) current solution point locates at the last sampling point and violates the second stopping criterion; (b) new sampling point is added to the adaptive surrogate model and doubles the upper bound of the total processing time.

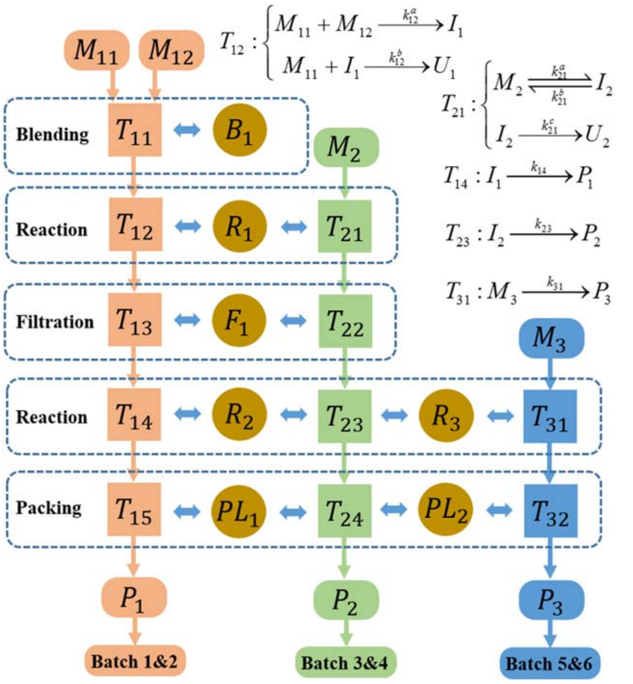
[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

```
$n \leftarrow 0$, $flag \leftarrow 1$
while ($flag = 1$) do
    Solving (Surrogate_MILP) to obtain a solution
    $flag \leftarrow 0$, $i \leftarrow 1$, $n \leftarrow n+1$
    while ($i < |I|+1$) do
        if ( the first stopping criterion for batch $i$ is not satisfied) then
            $flag \leftarrow 1$
            Add new sampling point at the location of the solution point
            Build a new surrogate model
        end if
        if (the second stopping criterion for batch $i$ is not satisfied) then
            $flag \leftarrow 1$
            Add a new sampling point that doubles the total processing time
            Build a new surrogate model
        end if
        $i \leftarrow i+1$
    end while
end while
```

Figure 7. Pseudocode for adaptive surrogate-based algorithm.

In this work, all optimization problems are modeled in GAMS 24.4.1[77] and solved in a computer with Intel(R) Core (TM) i5-2400 CPU @ 3.10 GHz, 8-GB RAM.

Sequential batch model



$T_{12}: \begin{cases} M_{11} + M_{12} \xrightarrow{k_{12}^a} I_1 \\ M_{11} + I_1 \xrightarrow{k_{12}^b} U_1 \end{cases}$
$T_{21}: \begin{cases} M_2 \underset{k_{21}^b}{\overset{k_{21}^a}{\rightleftharpoons}} I_2 \\ I_2 \xrightarrow{k_{21}^c} U_2 \end{cases}$
$T_{14}: I_1 \xrightarrow{k_{14}} P_1$
$T_{23}: I_2 \xrightarrow{k_{23}} P_2$
$T_{31}: M_3 \xrightarrow{k_{31}} P_3$
M_{11}
M_{12}
Blending
T_{11}
B_1
M_2
Reaction
T_{12}
R_1
T_{21}
Filtration
T_{13}
F_1
T_{22}
M_3
Reaction
T_{14}
R_2
T_{23}
R_3
T_{31}
Packing
T_{15}
PL_1
T_{24}
PL_2
T_{32}
P_1
P_2
P_3
Batch 1&2
Batch 3&4
Batch 5&6

Figure 8. Resource task network for sequential batch process scheduling.

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

The resource task network for the case study is presented in Figure 8. In this sequential batch model, there are three production lines which have different production stages. For the first product P_1, the first task T_{11} is blending the two raw materials M_{11} and M_{12} together. The second task T_{12} contains two chemical reactions. All the corresponding chemical reaction equations are also given in Figure 8. The third task T_{13} is a filtration process which is designed to filter out all unwanted materials. The fourth task T_{14} is a first-order reaction. The last task T_{15} is a packing process which always appears at the end of a production line. Parallel units are used in the batch production process of this work. As shown in Figure 8, reactor R_2 is the same as R_3 and packing line PL_1 is identical to packing line PL_2. Consequently, the fourth task for P_1 can be conducted either in R_2 or in R_3. Compared with P_1, the second product P_2 does not have a blending task. The third product P_3 only has a first-order reaction task and a packing task.

Among all the operational stages for three products, only the chemical reactions have flexible recipes, while other tasks, such as blending tasks, filtration tasks, and packaging tasks, have fixed operational recipes. The dynamics of these chemical reactions are of great importance for improvement of the overall performance of the batch process. A general structure for the batch reactor is given in Figure 9. The reactions occur in batch reactors and they might be exothermic or endothermic depending on the corresponding heats of reaction. The batch reactor is covered by a jacket. Pipes for both the heating and cooling water streams are connected with the jacket. The temperature of the jacket can be manipulated by the input flow rates of heating and cooling water. Heat can be transferred between the batch reactor and the jacket. Thus, the temperature of the reactor can also be controlled by the manipulated variables, the input flow rates of the heating and cooling water.

The dynamics of one batch reaction can be described by the following differential and algebraic equations. The rate constant k_s for chemical reaction s is determined by the Arrhenius equation in Eq. 36, where E_s is the activation energy for reaction s, R is the universal gas constant, and T_R is the temperature of the reactor. The mass rate law, as shown in Eq. 37, determines the chemical reaction rate r_s, where v_{sp} denotes the stoichiometric coefficient in the corresponding reaction and C_p is concentration for material p. s denotes the index for different reactions. p is the index for different materials. n_p denotes the number of materials in the reaction

$$k_s = Z_s e^{-E_s/(R \cdot T_R)} \tag{36}$$

$$r_s = k_s \prod_{p=1}^{n_p} \left(C_p\right)^{v_{sp}} \tag{37}$$

Equation set (38) represents the mass balance equations, which are used to describe the change of concentration for different species. n_s is the number of reactions involved

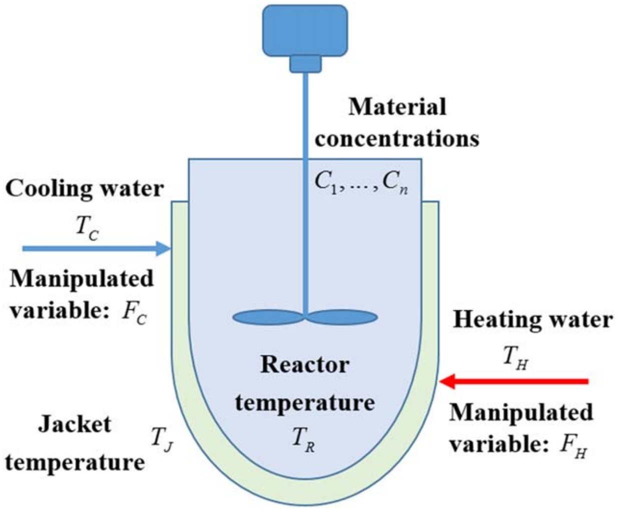


Figure 9. The general representation of a batch reaction.

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

$$
\begin{aligned}
\frac{d}{dt}C_1(t) &= \sum_{s=1}^{n_s} v_{s1} r_s \\
&\vdots \\
\frac{d}{dt}C_p(t) &= \sum_{s=1}^{n_s} v_{sp} r_s
\end{aligned}
\tag{38}
$$

Equation 39 is established according to the heat balance for a batch reactor, calculated from the reaction heat and the heat flux from the jacket. ΔH_s denotes the heat of reaction for reaction s. UA is the heat transfer coefficient between the reactor and the jacket. ρ_R denotes the density of reactor. c_R denotes the heat capacity of reactor. V_R is the volume of the reactor

$$
\frac{d}{dt}T_R(t) = \frac{\sum_{s=1}^{n_s} r_s(-\Delta H_s)}{\rho_R c_R} + \frac{UA(T_J - T_R)}{V_R \rho_R c_R} \tag{39}
$$

The temperature of the jacket is described by Eq. 40, which considers heat transfer with the heating and cooling water, and heat exchange with the batch reactor. T_J denotes the temperature of the jacket. F_H and T_H are the flow rate and the temperature of heating water. F_c and T_C denotes the flow rate and the temperature of cooling water. ρ_J denotes the density of jacket. c_J denotes the heat capacity of jacket. V_J is the volume of the jacket. The manipulated variables are the flow rate of the heating water and the flow rate of the cooling water.

$$
\frac{d}{dt}T_J(t) = \frac{F_H T_H + F_C T_C - (F_H + F_C) T_J}{V_J} + \frac{UA(T_R - T_J)}{V_J \rho_J c_J} \tag{40}
$$

Changing the flow rates of the input water will lead to altering the temperature of the jacket. Because there is heat transfer between the batch reactor and the jacket, the temperature of the reactor will change along with the temperature of the jacket. The temperature of the reactor has direct influence on the chemical reaction rates according to the Arrhenius equation. The reaction rates determine the dynamics of the chemical reactions. Thus, the manipulated variables can exert influence on the dynamics of the chemical reactions.

Multistage dynamic optimization

In this subsection, we present how to conduct an experimental design for generating data for the surrogate model. Particularly, a general approach to solve the multistage dynamic optimization problem for the sequential batch process is presented. We choose the dynamic optimization problem with the objective to minimize the total processing time of the first production line. Other dynamic optimization problems, such as minimizing the total processing cost at given total processing time, can be solved by following the same procedure. Dynamic optimization problems for other production lines can also follow the same procedure.

As shown in Figure 8, a production line may have a set of operational stages, including blending, reaction, filtration, and packing. Among those stages, only the stages of the reaction task can have a flexible recipe. Blending, filtration, and packing stages are not considered for dynamic optimization as they use fixed recipes. In the first production line, we only need to take the second stage and the fourth stage into consideration for the dynamic optimization problems. Namely, we need to determine the optimal dynamic trajectories for these two chemical reaction stages.

For multiobjective optimization, the collocation method[54] is applied to solve the dynamic optimization problem. We use 20 equal-length finite elements in each dynamic stage and 3 collocation points in every finite element. The length of the finite element in each stage is a variable, so the processing time of one reaction stage can vary according to different operating trajectories. Specifically, the Radau IIA method is adopted in this work.[10] The corresponding coefficients used in the collocation method are listed as follows

$$
a_{qq'} = \begin{bmatrix} \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{bmatrix},
$$

$$
c_q = \begin{bmatrix} \frac{4-\sqrt{6}}{10} \\ \frac{4+\sqrt{6}}{10} \\ 1 \end{bmatrix}, \text{ and } \quad b_q = \begin{bmatrix} \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{bmatrix}
$$

The multistage dynamic optimization problem to minimize the total processing time for P_1 is a nonlinear problem which is solved by CONOPT 3 with 0.64 CPUs. There are 1792 constraints and 1818 continuous variables in this problem. The optimal objective value of this problem is 2.289 h. We note that we also use one global optimal optimizer, BARON 14.4, to solve this multistage dynamic optimization problem. The same objective value 2.289 h is returned when the predetermined computational time limit of 50 h is reached, while the relative gap if this solution returned by BARON 14.4 is 47.53%.

The optimal trajectories for concentrations of different materials returned from the dynamic optimization are illustrated in Figure 10. In the second stage, M_{11} and M_{12} are reactants while I_1 is the product. In the fourth stage, I_1 is the reactant, and P_1 is the final product for the production line. The initial values for the second stage, including the initial concentrations for M_{11} and M_{12}, are fixed parameters. The initial values for the fourth stage are the final values of the second stage, the initial concentration of I_1 in the fourth stage is the same as the final concentration of I_1 in the second stage. We note that there is a quality constraint for the final product P_1 such that the concentration of P_1 should be greater than 4000 kmol/m^3.

The optimal trajectories for reactors' and jackets' temperatures are displayed in Figure 11. In each individual reaction stage, the temperature of the reactor and the temperature of the jacket are manipulated by the flow rate of the heating water and the flow rate of the cooling water. For each reaction, due to heat exchange with the ambient air during material transition, the initial temperature of the reactor should be equivalent to the environmental temperature. During the chemical reaction, the flow rate of the heating water is usually high at the beginning of the reaction. The heating water leads to temperature increase in the reactor, thus giving rise to a fast reaction. However, due to safety considerations, the temperature of the reactor should not go beyond the safety boundary throughout the production process. The safety boundary for the temperature of reactor R_1 is 350 K, while the safety

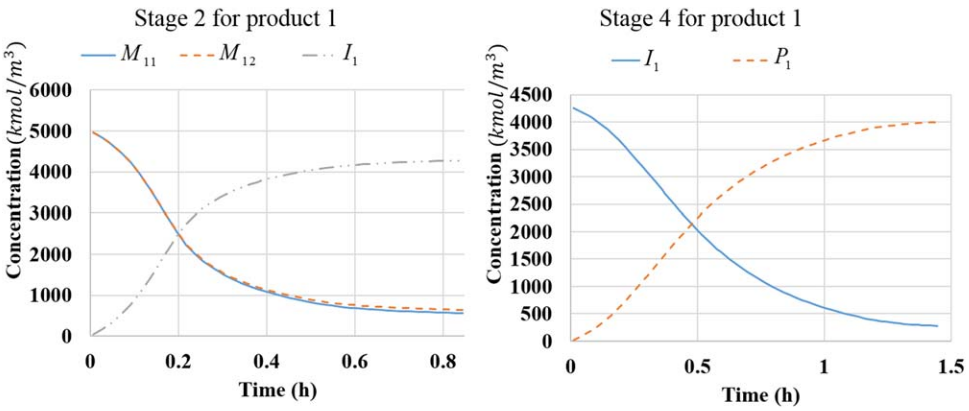


Figure 10. Concentration profile in Stage 2 and Stage 4 of Product 1.

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

boundary for the temperature of reactor R_{II} and R_{III} is 360 K. The safety boundary for the temperature of all jackets is set at 370 K. For each reaction stage, if the reaction process is nearly over, the temperature of the reactor should be controlled by cooling water to decrease the temperature to below a certain threshold. The reason lies in the fact that if the temperature of the products from the reaction is extremely high, they cannot be processed by the following operational stage. The threshold for the outlet temperature is set at 320 K.

The optimal trajectories for input heating water and cooling water for different stages in the first production line are displayed in Figure 12. The flow rate of the heating water is usually high at the beginning of the reaction to enhance the reaction rate, while in general the flow rate of the cooling water is high toward the end of the reaction to force the temperature of the reactor below the temperature threshold. We note that there are utility constraints for the flow rate of input water. The sum of the heating water flow rate and the cooling water flow rate should be less than 20 m^3/h for the reaction stages conducted in reactor R_{I}, and the sum of the flow rates for the heating and cooling water should be less than 30 m^3/h for the reactions conducted in reactor R_{II} or R_{III}.

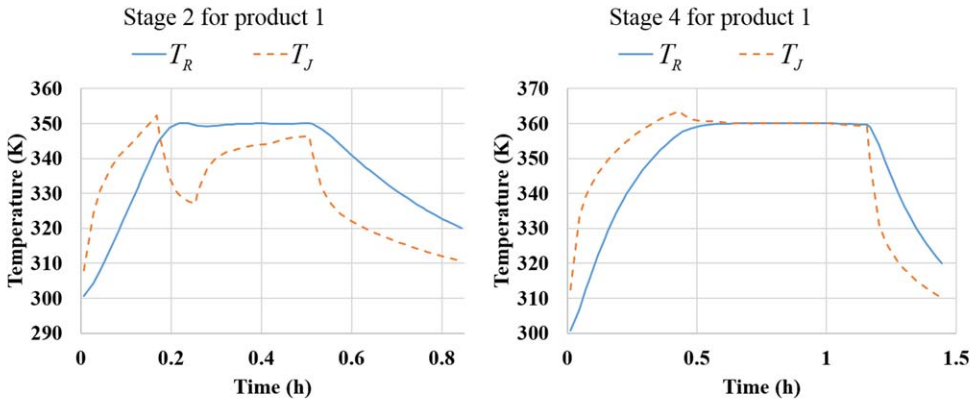


Figure 11. Temperature profile in Stage 2 and Stage 4 of Product 1.

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

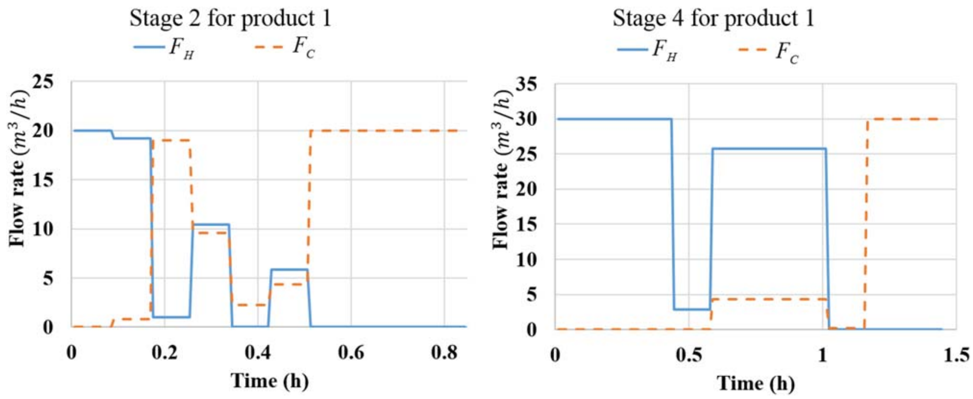


Figure 12. Profiles of input water in Stage 2 and Stage 4 of Product 1.

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

Table 1. Model and Solution Statistics for Different Methods

	Adaptive Surrogate[a]	Sequential	Full Space
Type	MILP	MILP	MINLP
Bin. Var.	210	188	188
Con. Var.	4109	371	9145
Constraints	4474	614	9262
Optimizer	CPLEX 12	CPLEX 12	SBB
Max Obj.($)	4937.88	3776.26	4505.95[b]
CPU (s)	24.27	0.25	360,000
Iteration	6	NA	NA

[a] For adaptive surrogate modeling-based algorithm, the statistics is based on the model and solution in the last iteration.

[b] This is the local optimal solution returned by the full space method. This solution is first found at 145,566 nodes when the computational time is equal to 317,392 s (about 88.1 h). The computational time limit is 100 h.

Comparison of different methods

In this case study, we first solve the integrated scheduling and dynamic optimization problem with the adaptive surrogate modeling-based algorithm. To demonstrate the performance of our proposed method, we also apply the conventional sequential method and the full space simultaneous method to solve this sequential batch case study. The model and solution statistics for these methods are given in Table 1. For the adaptive surrogate modeling-based algorithm, there are 210 binary variables, 4109 continuous variables, and 4474 constraints in this MILP problem, which is solved by CPLEX 12 within 24.27 CPUs. This objective is to maximize the total profit across the scheduling horizon for this sequential batch process. The maximum profit returned by the adaptive surrogate modeling-based algorithm is $4937.88. Our solution method takes six iterations to satisfy all the stopping criterions and terminate.

The adaptive surrogate model for each batch is shown in Figure 13. All of the adaptive surrogate models start with only two sampling point. From the first iteration to the fifth iteration, there are some solution points that do not satisfy the stopping criterions. Thus, new sampling points for these batches are added into their surrogate models adaptively according to the processing times of the current solutions. As we can see in Figure 13, the solution point of Batch 2 never violates the stopping criterions in any iteration. Thus, the final adaptive surrogate model for Batch 2 only has two sampling points. In contrast, the solution point for Batch 1 violates the stopping criterions in every iteration, and there are seven sampling points in the final adaptive surrogate model for Batch 1. We note that most of the solution points are located exactly on the existing sampling points. Only the solution point for Batch 1 is not located exactly on any of the existing sampling points. The predetermined tolerance for Batch 1 is 0.001, which means that the absolute error between the solution point and the nearest sampling point is only one thousandth of the time interval between the two adjacent sampling points next to the solution point. As we can see in Figure 13, the absolute error between the solution points and the nearest sampling point for Batch 1 should be less than 3.6 s, which is about four orders of magnitude smaller compared with the upper bound of the scheduling horizon (9 h).

The profit returned by the adaptive surrogate-based method in each iteration is shown in Figure 14. The profit improves significantly in the first three iterations while it increases slowly in the last several iterations. Specifically, for the solution point in the first iteration, the sales of all the products are $12,027.6, the cost for the production process is $7622.7, and the profit is $4404.9. For the solution point in the second iteration, the profit of $4804.2 is calculated by subtracting the costs

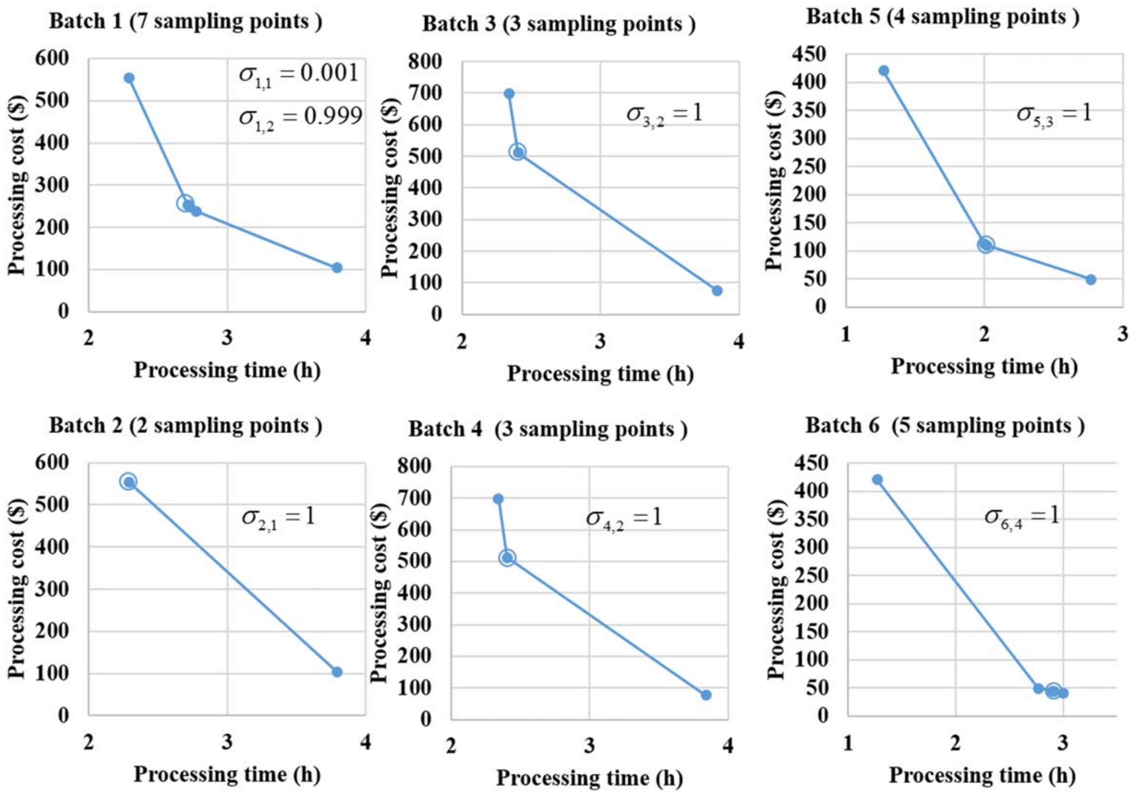


Figure 13. Adaptive surrogate model for each batch.

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

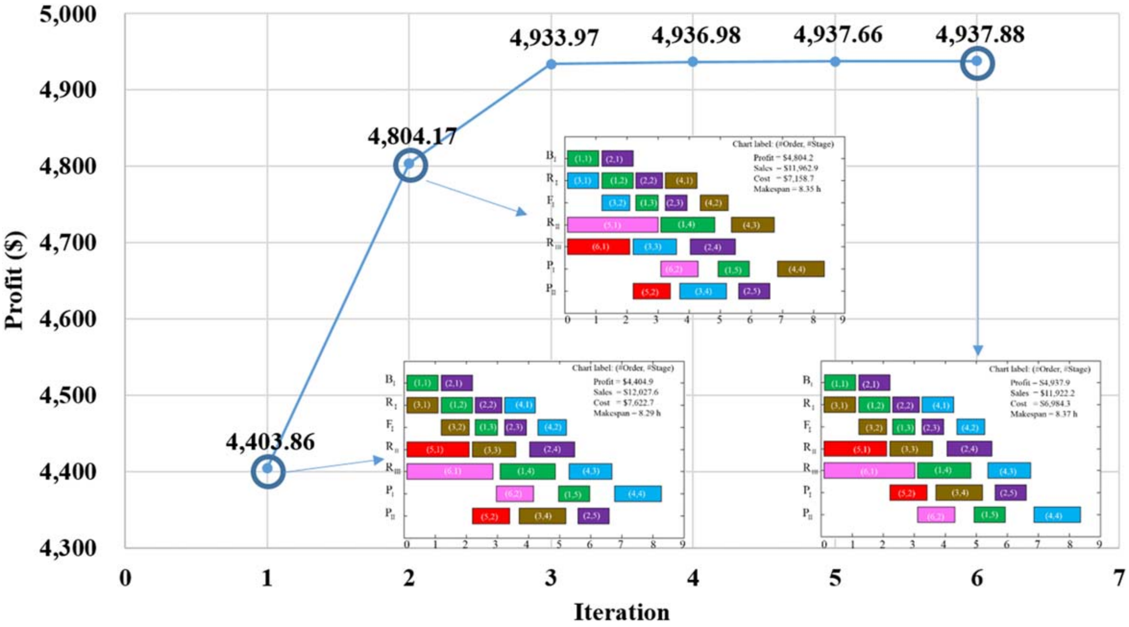


Figure 14. Profit in each iteration for adaptive surrogate-based method.

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

$7158.7 from the sales $11,962.9. The profit of the last solution iteration is $4937.9. The makespan of 8.35 h in the second iteration is the same as that in the last iteration and is larger than the makespan in the first iteration, which is 8.29 h. In addition to the variations in profit and makespan, we can also see scheduling differences between the iterations. In the first and last iteration batch five is delivered before batch six, while Batch 6 is delivered first in the second iteration. The last stage of batch four is manufactured in P_{I} in the first and second iteration while this stage is moved to P_{II} in the last iteration.

The Gantt chart of the production schedule returned by the adaptive surrogate-based approach is shown in Figure 15. As we can see, the sales of all products are $11,922.2 and the production process cost is $6984.3. The makespan returned by the adaptive surrogate modeling-based algorithm is 8.37 h.

For the conventional sequential method, the operating recipes in the scheduling problem are based on the fixed parameters provided by the dynamic optimization problems. The dynamic optimization problems and the scheduling problem are solved sequentially. As the dynamic optimization problems are solved independently without considering any information from the scheduling problem, the dynamic optimization is usually solved with the objective to some local criterions. In this work, the local criterion to be minimized is the total processing time. We formulate an MILP problem with sequential solution method, and there are 188 binary variables, 317 continuous variables, and 614 constraints in this problem. The objective is to maximize the total profit, and the maximum profit returned by the sequential method is $3776.26, solved by CPLEX 12 within 0.25 CPUs. The problem size of the sequential method is smaller than the adaptive surrogate modeling-based method. This is because the adaptive surrogate modeling-based method uses a large number of variables and constraints in the construction of the piecewise linear surrogate model. The adaptive surrogate modeling-based method uses more computational time than the sequential approach. However, the adaptive surrogate modeling-based algorithm returns 30.76% higher profit than the sequential method.

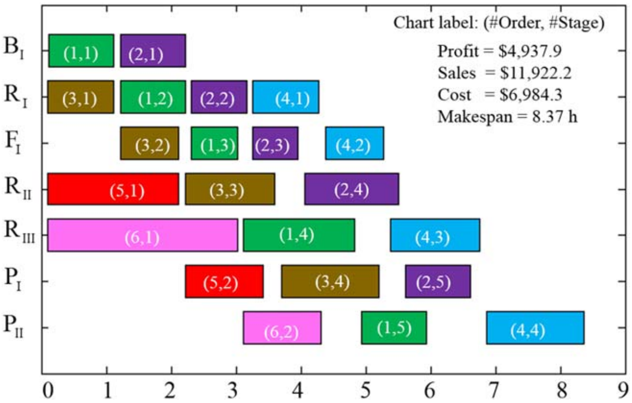


Figure 15. Production scheduling by adaptive surrogate modeling-based method.

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

The Gantt chart of the batch production scheduling provided by the sequential approach is shown in Figure 16. The makespan by sequential approach is 7.69 h, which is shorter than that provided by the adaptive surrogate-based approach. The solution agrees with the local criterion to minimize the total processing time, and the sequential method aims to finish each batch as soon as possible.

The third solution approach to solve the integrated scheduling and dynamic optimization problem is the full space simultaneous method.[78] The simultaneous method solves the monolithic model of the integrated scheduling and dynamic optimization problem directly. For the differential and algebraic equations in the dynamic optimization, we use the same collocation method discussed in multistage dynamic optimization subsection. That is, we adopt the Radau IIA method, discretize each dynamic stage into 20 finite elements, and choose 3 collocation points for each element. For the simultaneous method, there are 188 binary variables, 9145 continuous

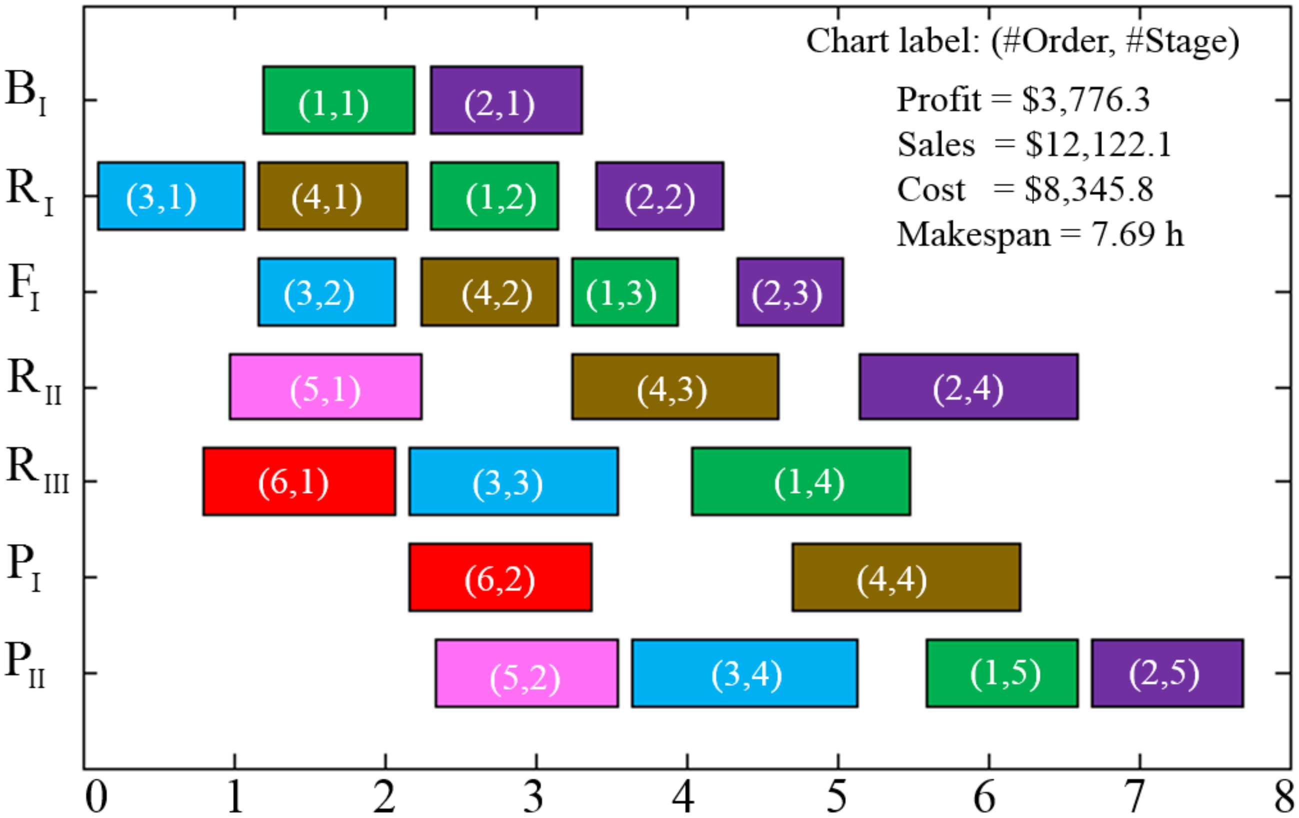


Figure 16. Production scheduling by sequential method.

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

variables, and 9262 constraints in this MINLP problem. The batch production scheduling provided by full space simultaneous method is shown in Figure 17. The problem size by the simultaneous method is much larger than the other two methods. The optimizer employed for the sequential method is Simple Branch and Bound (SBB), which is a commonly used optimizer for integrated scheduling and dynamic optimization problems.[12,13,17] The objective returned by the simultaneous method is $4505.9 when the optimizer reaches the computational time limit 360,000 CPUs, that is, 100 h. We note that BARON 14.4 fails to return a feasible solution for this problem. The computational time of the full space simultaneous method is four orders of magnitude more than the adaptive surrogate modeling-based method, and our proposed method returns a 9.59% higher profit than the simultaneous method. We note that we solve the MINLP problem without assigning any initial point. The solution returned by the simultaneous method is a local optimal solution, which is first found at 145,566 in 317,392 CPUs (about 88.1 h). The relative gap returned by the optimizer is 51.73% when the computational time reaches its limit of 100 h. The computational time used to find one feasible solution for the simultaneous approach is about 10 times that of the scheduling horizon.

To analyze the importance of the initial points, we set the binary assignment variables returned by the sequential method

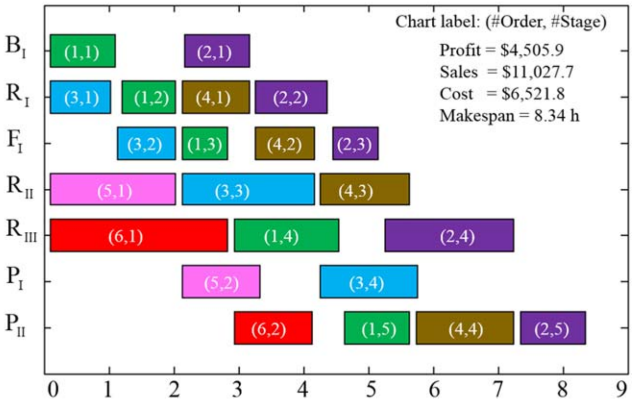


Figure 17. Production scheduling by full space simultaneous method.

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

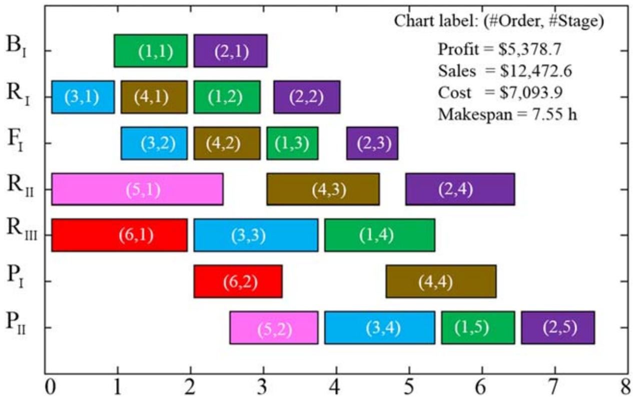


Figure 18. Production scheduling by full space simultaneous method initialized by the sequential method.

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

as the initial points and solve the problem again by the full space simultaneous method. The Gantt chart for this problem is shown in Figure 18. The optimizer returns a local optimal solution \$5387.7 when the computational limit 3600 CPUs is reached. The final relative gap returned by the optimizer is 43.35%. This solution is first obtained at about 65 CPUs. The full space simultaneous method return a 9.1% higher profit than our proposed method if the simultaneous method has a good initial points. However, in most cases, a good initial point and a reasonable computational time limit cannot be determined in advance. Thus, with the full space simultaneous method, a good solution with a short computational time is difficult to obtain in practice. The full space simultaneous solution method for the integrated scheduling and dynamic optimization problem is usually large scale and difficult to solve. Consequently, the full space simultaneous method is not recommended in this problem.

Conclusion

In this work, we proposed a novel adaptive surrogate modeling-based algorithm for the integrated scheduling and dynamic optimization problem for sequential batch processes. We formulated the integrated optimization problem as an MIDO problem, which was then reformulated into a large scale MINLP problem by means of the orthogonal collocation method. The monolithic model of the integrated optimization problem was challenging to be solved directly. We took advantage of the bilevel structure of the integrated optimization problem, and proposed an efficient solution algorithm based on adaptive surrogate modeling. Because the dynamic optimization problems were linked with the scheduling problem only via the processing times and processing costs, we were able to replace the dynamic optimization problems with a set of surrogate models based on piecewise linear functions. The surrogate models were then updated adaptively by adding new sampling points to the current surrogate model or by doubling the upper bound of the total processing time.

We used a case study of a multiproduct sequential batch process to demonstrate the applicability of our adaptive surrogate modeling-based algorithm. The sequential batch process had seven units for three production lines. Each production line had three to five operational tasks. The objective was to maximize the total profit. The results showed that the sequential method used 0.25 CPUs to return a profit of \$3776.3, while the proposed method returned a 31% higher profit of \$4939.9 within 24.27 CPUs. Compared with the full space simultaneous method, which returned a profit of \$4505.9 within 360,000 CPUs, our proposed method used less computational time by more than four orders of magnitude and returned a 9.59% higher profit.

In the case of highly nonlinear models such as batch distillation and batch reactor separator networks with potential recycles,[20,21] the solution of the adaptive surrogate model requires relatively high computational efforts. Sequential batch processes, in which operations of batch splitting, mixing, and resizing are not allowed, are the focus of this work. Surrogate models for dynamic models in sequential batch processes only have two dimensions, namely the processing cost is expressed as a function of the processing time. However, in more sophisticated cases such as complex batch processes with general network structure, where batch splitting, mixing, and resizing are allowed, their surrogate models would involve multiple dimensions, for example, the processing cost could be a function of the processing time, the batch size and the concentrations of input materials. The multidimensional surrogate modeling and optimization for this problem leads to substantial computational challenges and requires more powerful and efficient solution methods. This would be a direction of our extension of this work.

Notation

Indices

i = batch
l = stage
j = unit
k = slot
m = sampling point
n = iteration
p = material in chemical reaction
r = finite element
s = chemical reaction
q = collocation point

Sets

IL_j = batch stages, indexed by (i,l), which can be processed by unit j
J_{il} = units capable for processing batch i at stage l
K_j = time slots for unit j
L_i = operational stages for batch i
R_i^n = index set of the discrete points for batch i in nth iteration

Parameters

$a_{qq'}$ = collocation matrix
b_q = collocation vector
c_q = collocation vector
c^{Fix} = total fixed cost
c_i^{Fix} = fixed cost for batch i
c_i^P = price for batch i
d_i^{I} = first due date function for batch i
d_i^{II} = second due date function for batch i
n_r = number of finite elements
n_q = number off collocation points
n_p = number of materials in the reaction
n_s = number of reactions involved
st_{il} = setup time of stage l for order i
t_H = upper bound of scheduling horizon
x_i^0 = initial condition for batch i

Greek letters

ε_i = ε value of multiobjective optimization for batch i
$\varepsilon_i^{(m)}$ = ε value of the mth data point in multiobjective optimization for batch i
$\varepsilon_i^{\min}$ = ε value of multiobjective optimization for batch i
$\Delta\varepsilon_i$ = ε value of multiobjective optimization for batch i
η_i = optimal-value function of dynamic optimization for batch i
φ_{il} = processing cost in dynamic model of stage l for batch i
σ_{im} = SOS2 variables in piecewise linear function
τ_{il} = ending time in dynamic model of stage l for batch i
Δ_{il} = length of finite element in dynamic stage l of batch i

Binary variables

ζ_{im} = SOS2 variables in piecewise linear Pareto curve
OW_i^{I} = indicator of first segment in batch price function for batch i
OW_i^{II} = indicator of second segment in batch price function for batch i
W_{ijkl} = assignment of operational stage l of batch i to time slot k of unit j

Continuous variables

C_i^{Var} = variable cost for batch i
$Cost$ = total production cost
DT_i^{I} = first component of delivery date for batch i
DT_i^{II} = second component of delivery date for batch i

DT_i^{III} = second component of delivery date for batch i

ODT_i = batch delivery date of batch i

P_i^C = total processing cost for batch i

$P_i^{C(m)}$ = total processing cost of the mth data point for batch i

PC_{il} = processing cost of stage l for batch i

$Profit$ = production profit

PR_i = final batch price for batch i

PT_{il} = processing time of stage l for batch i

$PT_{il}^{(m)}$ = processing time for the mth data point of stage l for batch i

S_{jk} = slack variables for slot k in unit j

$Sales$ = sales of all batches

T_{il} = time variable in dynamic model of stage l for batch i

T_{il}^{rq} = discretized time variable for stage l of batch i

TEI_{il} = ending time of stage l for batch i

TEJ_{jk} = ending time of slot k in unit j

TSI_{il} = starting time of stage l for batch i

TSJ_{jk} = starting time of slot k in unit j

U_{il} = inputs in dynamic model of stage l for batch i

U_{il}^{rq} = discretized time variable in dynamic model of stage l for batch i

UT_{ijkl} = variable used for linearizing $WT_{ijkl}\mathrm{PT}_{il}$

V^{Sch} = collection of variables in scheduling problem

V_i^{Dyn} = collection of variables in dynamic model for batch i

WT_{ijkl} = nearizing $WT_{ijkl} \cdot \mathrm{PT}_{il}$

X_{il} = state variables in dynamic model of stage l for batch i

X_{il}^{rq} = discretized state variables in dynamic model of stage l for batch i

X_i^f = final value of state variable for batch i

Y_{il} = outputs in dynamic model of stage l for batch i

Y_{il}^{rq} = discretized output variables in dynamic model of stage l for batch i

Literature Cited

1. Grossmann IE. Enterprise-wide optimization: a new frontier in process systems engineering. *AIChE J.* 2005;51:1846–1857.
2. Varma VA, Reklaitis GV, Blau G, Pekny JF. Enterprise-wide modeling & optimization—an overview of emerging research challenges and opportunities. *Comput Chem Eng.* 2007;31:692–711.
3. Grossmann IE. Advances in mathematical programming models for enterprise-wide optimization. *Comput Chem Eng.* 2012;47:2–18.
4. Chu Y, You F. Model-based integration of control and operations: overview, challenges, advances, and opportunities. *Comput Chem Eng.* In press. Doi: 10.1016/j.compchemeng.2015.1004.1011.
5. Baldea M, Harjunkoski I. Integrated production scheduling and process control: a systematic review. *Comput Chem Eng.* 2014;71:377–390.
6. Wassick JM, Agarwal A, Akiya N, Ferrio J, Bury S, You F. Addressing the operational challenges in the development, manufacture, and supply of advanced materials and performance products. *Comput Chem Eng.* 2012;47:157–169.
7. 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.
8. 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.
9. 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.
10. Chu Y, You F. Integrated scheduling and dynamic optimization of complex batch processes with general network structure using a generalized benders decomposition approach. *Ind Eng Chem Res.* 2013; 52:7867–7885.
11. Chu Y, You F, Wassick JM, Agarwal A. Integrated planning and scheduling under production uncertainties: bi-level model formulation and hybrid solution method. *Comput Chem Eng.* 2015;72:255–272.
12. Flores-Tlacuahuac A, Grossmann IE. Simultaneous cyclic scheduling and control of a multiproduct CSTR. *Ind Eng Chem Res.* 2006;45: 6698–6712.
13. Flores-Tlacuahuac A, Grossmann IE. Simultaneous scheduling and control of multiproduct continuous parallel lines. *Ind Eng Chem Res.* 2010;49:7909–7921.
14. 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.
15. Chu Y, You F. Integrated scheduling and dynamic optimization by Stackelberg game: bilevel model formulation and efficient solution algorithm. *Ind Eng Chem Res.* 2014;53:5564–5581.
16. Chu Y, You F. Integrated planning, scheduling, and dynamic optimization for batch processes: MINLP model formulation and efficient solution methods via surrogate modeling. *Ind Eng Chem Res.* 2014; 53:13391–13411.
17. Zhuge J, Ierapetritou MG. Integration of scheduling and control with closed loop implementation. *Ind Eng Chem Res.* 2012;51:8550–8565.
18. Chu Y, You F. Moving horizon approach of integrating scheduling and control for sequential batch processes. *AIChE J.* 2014;60:1654–1671.
19. Papageorgiou LG, Charalambides MS, Shah N, Pantelides CC. Optimal operation of thermally coupled batch processes. In *4th European Symposium on Computer Aided Process Engineering*. INST Chemical Engineers: Doublin, Ireland, 1994:71–78.
20. Charalambides M, Shah N, Pantelides C. Synthesis of batch reaction/distillation processes using detailed dynamic models. *Comput Chem Eng.* 1995;19:167–174.
21. Sharif M, Shah N, Pantelides C. On the design of multicomponent batch distillation columns. *Comput Chem Eng.* 1998;22:S69–S76.
22. Sharif M, Shah N, Pantelides C. Design of integrated batch processes with discrete and continuous equipment sizes. *Comput Chem Eng.* 1999;23:S117–S120.
23. Chu Y, You F. Integrated scheduling and dynamic optimization of sequential batch processes with online implementation. *AIChE J.* 2013;59:2379–2406.
24. Nie Y, Biegler LT, Villa CM, Wassick JM. Discrete time formulation for the integration of scheduling and dynamic optimization. *Ind Eng Chem Res.* 2014;54(16):4303–4315.
25. Nie YS, Biegler LT, Wassick JM. Integrated scheduling and dynamic optimization of batch processes using state equipment networks. *AIChE J.* 2012;58:3416–3432.
26. Floudas CA, Lin X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comput Chem Eng.* 2004;28:2109–2129.
27. Méndez CA, Cerdá J, Grossmann IE, Harjunkoski I, Fahl M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput Chem Eng.* 2006;30:913–946.
28. 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.
29. Chu Y, Wassick JM, You F. Efficient scheduling method of complex batch processes with general network structure via agent-based modeling. *AIChE J.* 2013;59:2884–2906.
30. Allgor R, Barton P. Mixed-integer dynamic optimization I: problem formulation. *Comput Chem Eng.* 1999;23:567–584.
31. Bansal V, Sakizlis V, Ross R, Perkins JD, Pistikopoulos EN. New algorithms for mixed-integer dynamic optimization. *Comput Chem Eng.* 2003;27:647–668.
32. Biegler L. Technology advances for dynamic real-time optimization. *Comput Aided Chem Eng.* 2009;27:1–6.
33. Biegler LT. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes, Vol 10*. SIAM, Philadelphia, Pennsylvania, USA 2010.
34. 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.
35. 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.
36. Shi H, Chu Y, You F. Novel optimization model and efficient solution method for integrating dynamic optimization with process operations of continuous manufacturing processes. *Ind Eng Chem Res.* 2015;54:2167–2187.
37. Keesman KJ. Application of flexible recipes for model building, batch process optimization and control. *AIChE J.* 1993;39:581–588.
38. Ferrer-Nadal S, Méndez CA, Graells M, Puigjaner L. Optimal reactive scheduling of manufacturing plants with flexible batch recipes. *Ind Eng Chem Res.* 2007;46:6273–6283.
39. Ferrer-Nadal S, Méndez CA, Graells M, Puigjaner L. A mathematical programming approach including flexible recipes to batch operation rescheduling. *Comput Aided Chem Eng.* 2006;21:1377–1382.
40. Capón-García E, Moreno-Benito M, Espuna A. Improved short-term batch scheduling flexibility using variable recipes. *Ind Eng Chem Res.* 2011;50:4983–4992.

41. Wang GG, Shan S. Review of metamodeling techniques in support of engineering design optimization. *J Mech Des.* 2007;129:370–380.
42. Burnham AJ, Viveros R, MacGregor JF. Frameworks for latent variable multivariate regression. *J Chemom.* 1996;10:31–45.
43. Forrester AI, Keane AJ. Recent advances in surrogate-based optimization. *Prog Aerospace Sci.* 2009;45:50–79.
44. Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK. Surrogate-based analysis and optimization. *Prog Aerospace Sci.* 2005;41:1–28.
45. Box GE, Draper NR. *Empirical Model-Building and Response Surfaces*. Wiley, New York, USA, 1987.
46. Forrester A, Sobester A, Keane A. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2008.
47. Myers RH, Montgomery DC, Anderson-Cook CM. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments, Vol. 705*. Wiley, Hoboken, New Jersey, USA 2009.
48. Koziel S, Leifsson L. Surrogate-based modeling and optimization. *Appl Eng*. Springer, New York, USA, 2013.
49. Reklaitis GV. Overview of scheduling and planning of batch process operations. *Batch Processing Systems Engineering*. Springer, Berlin, Germany 1996:660–705.
50. Potts CN, Kovalyov MY. Scheduling with batching: a review. *Eur J Oper Res.* 2000;120:228–249.
51. Wassick JM, Ferrio J. Extending the resource task network for industrial applications. *Comput Chem Eng.* 2011;35:2124–2140.
52. Bhatia T, Biegler LT. Dynamic optimization in the design and scheduling of multiproduct batch plants. *Ind Eng Chem Res.* 1996;35:2234–2246.
53. Pinto JM, Grossmann IE. A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Ind Eng Chem Res.* 1995;34:3037–3051.
54. Cuthrell JE, Biegler LT. On the optimization of differential-algebraic process systems. *AIChE J.* 1987;33:1257–1270.
55. Biegler LT, Cervantes AM, Wächter A. Advances in simultaneous strategies for dynamic process optimization. *Chem Eng Sci.* 2002;57:575–593.
56. Biegler LT. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Comput Chem Eng.* 1984;8:243–247.
57. Capón-García E, Guillén-Gosálbez G, Espuña A. Integrating process dynamics within batch process scheduling via mixed-integer dynamic optimization. *Chem Eng Sci.* 2013;102:139–150.
58. Cozad A, Sahinidis NV, Miller DC. Learning surrogate models for simulation-based optimization. *AIChE J.* 2014;60:2211–2227.
59. Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. *J Global Optim.* 1998;13:455–492.
60. Gorissen D, Couckuyt I, Demeester P, Dhaene T, Crombecq K. A surrogate modeling and adaptive sampling toolbox for computer based design. *J Mach Learn Res.* 2010;11:2051–2055.
61. Lophaven S, Nielsen H, Sondergaard J. *DACE–A MATLAB Kriging Toolbox. Technical Report No. IMM-TR-2002-12*. Kongens Lyngby: Technical University of Denmark, 2002.
62. Loose J-P. *Physical and Surrogate Modeling for Complex Manufacturing Process Design and Control*. ProQuest, Ann arbor, Michigan, USA 2008.
63. Conn AR, Scheinberg K, Vicente LN. *Introduction to Derivative-Free Optimization, Vol. 8*. SIAM, Philadelphia, Pennsylvania, USA 2009.
64. Koziel S, Yang X-S. *Computational Optimization, Methods and Algorithms, Vol. 356*. Springer Science & Business Media, 2011.
65. Biegler LT, Lang Y-d, Lin W. Multi-scale optimization for process systems engineering. *Comput Chem Eng.* 2014;60:17–30.
66. Croxton KL, Gendron B, Magnanti TL. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Manag Sci.* 2003;49:1268–1273.
67. Holmberg K. Solving the staircase cost facility location problem with decomposition and piecewise linearization. *Eur J Oper Res.* 1994;75:41–61.
68. Croxton KL, Gendron B, Magnanti TL. Variable disaggregation in network flow problems with piecewise linear costs. *Oper Res.* 2007;55:146–157.
69. Aghezzaf EH, Wolsey LA. Modelling piecewise linear concave costs in a tree partitioning problem. *Discrete Appl Math.* 1994;50:101–109.
70. Glover F. Improved linear integer programming formulations of nonlinear integer problems. *Manag Sci.* 1975;22:455–460.
71. You F, Pinto JM, Grossmann IE, Megan L. Optimal distribution-inventory planning of industrial gases. II. MINLP models and algorithms for stochastic cases. *Ind Eng Chem Res.* 2011;50:2928–2945.
72. You F, Grossmann IE. Stochastic inventory management for tactical process planning under uncertainties: MINLP models and algorithms. *AIChE J.* 2011;57:1250–1277.
73. Yue D, You F. Game-theoretic modeling and optimization of multi-echelon supply chain design and operation under Stackelberg game and market equilibrium. *Comput Chem Eng.* 2014;71:347–361.
74. Gong J, You F. Optimal design and synthesis of algal biorefinery processes for biological carbon sequestration and utilization with zero direct greenhouse gas emissions: MINLP model and global optimization algorithm. *Ind Eng Chem Res.* 2014;53:1563–1579.
75. Pantelides CC. Unified frameworks for optimal process planning and scheduling. *Proceedings on the Second Conference on Foundations of Computer Aided Operations*. Cache Publications: New York, USA, 1994.
76. Biegler LT. An overview of simultaneous strategies for dynamic optimization. *Chem Eng Process.* 2007;46:1043–1053.
77. Richard E. *GAMS-A User's Guide*. Washington, DC: GAMS Development Corporation, 2013.
78. Cuthrell JE, Biegler LT. Simultaneous optimization and solution methods for batch reactor control profiles. *Comput Chem Eng.* 1989; 13:49–62.

Appendix A: Scheduling Problem of Sequential Batch Process

The entire time-slot scheduling model[53] can be divided into four sections: unit allocation section, timing matching section, batch price function section, and sales and cost section. The objective function is also given at the end.

1. Unit allocation

Constraint (A1) means that a batch is processed only once throughout the whole process

$$\sum_{j\in J_{il}}\sum_{k\in K_j} W_{ijkl}=1, \forall i, l \in L_i \tag{A1}$$

Constraint (A2) ensures no more than one stage of a batch is allocated to one time slot of a unit at a time. W_{ijkl} is the assignment binary variable. S_{jk} denotes the slack variable. All stages of batch i is included in set L_i. All time slots in unit j is included in set K_j. Set IL_j contains all pairs of batch-stage indexed by (i,l) which unit j can produce

$$\sum_{(i,l)\in IL_j} W_{ijkl}+S_{jk}=1, \forall j, k \in K_j \tag{A2}$$

Constraint (A3) are used to place all the occupied time slots to the beginning

$$S_{jk-1} \le S_{jk}, \forall j, k \in K_j \text{ and } k > 2 \tag{A3}$$

Constraint (A4) states that the slack variables are nonnegative

$$S_{jk} \ge 0, \forall j, k \in K_j \tag{A4}$$

2. Timing relation

Constraint (A5) means the ending time TEI_{il} of stage l for batch i is the sum of the starting time TSI_{il}, the setup time st_{il}, and the production time PT_{il}

$$TEI_{il}=TSI_{il}+st_{il}+PT_{il}, \forall i, l \in L_i \tag{A5}$$

Constraint (A6) ensures the ending time of the current stage $(l-1)$ is before the starting time of the next stage l for the same batch i

$$TSI_{il} \ge TEI_{i(l-1)}, \forall i, l \in L_i, l \ge 2 \tag{A6}$$

Constraint (A7) means the ending time TEJ_{jk} of time slot k for unit j is the sum of the starting time TSJ_{jk}, the setup time st_{il}, and the production time PT_{il}

$$TEJ_{jk}=TSJ_{jk}+\sum_{(i,l)\in IL_j} W_{ijkl}\cdot(st_{il}+PT_{il}), \forall j, k \in K_j \tag{A7}$$

Constraint (A8) ensures the ending time of the current time slot ($k-1$) is before the starting time of the next time slot k for the same unit j

$$TSJ_{jk} \geq TEJ_{j(k-1)}, \forall j, k \in K_j, k \geq 2 \tag{A8}$$

Constraints (A9) and (A10) ensure that the starting time of stage l of batch i is the same as the starting time of time slot k of unit j if the stage l is assigned to the time slot k. t_H is the upper bound of the scheduling horizon

$$TSI_{il}-TSJ_{jk} \geq -t_H \cdot \left(1-W_{ijkl}\right), \forall i, j \in J_{il}, k \in K_j, l \in L_i \tag{A9}$$

$$TSI_{il}-TSJ_{jk} \leq t_H \cdot \left(1-W_{ijkl}\right), \forall i, j \in J_{il}, k \in K_j, l \in L_i \tag{A10}$$

3. Batch price functions

In Figure 2, there are two due dates for batch i, d_i^{I} and d_i^{II}. The batch price is c_i^P if it is delivered before d_i^{I}. The batch price decreases with respect to time linearly if it is delivered after d_i^{I} and before d_i^{II}. The batch price is zero if it is delivered after d_i^{II}. The scheduling horizon t_H is the hard deadline for delivery date ODT_i.

Constraint (A11) means delivery date ODT_i is equal to the ending time of the last stage of batch i

$$ODT_i=TEI_{i(l=|L_i|)}, \forall i \tag{A11}$$

Constraint (A12) means the delivery date is the sum of three components DT_i^{I}, DT_i^{II}, and DT_i^{III}

$$ODT_i=DT_i^{\mathrm{I}}+DT_i^{\mathrm{II}}+DT_i^{\mathrm{III}} \tag{A12}$$

The binary variables OW_i^{I} and OW_i^{II} are used to express the location of the delivery date. For each batch, the relationship between the three binary variables and the three components of the delivery date is expressed by constraints (A13)–(A15)

$$OW_i^{\mathrm{I}} \cdot d_i^{\mathrm{I}} \leq DT_i^{\mathrm{I}} \leq d_i^{\mathrm{I}}, \forall i \tag{A13}$$

$$OW_i^{\mathrm{II}} \cdot \left(d_i^{\mathrm{II}}-d_i^{\mathrm{I}}\right) \leq DT_i^{\mathrm{II}} \leq OW_i^{\mathrm{I}} \cdot \left(d_i^{\mathrm{II}}-d_i^{\mathrm{I}}\right), \forall i \tag{A14}$$

$$DT_i^{\mathrm{III}} \leq OW_i^{\mathrm{II}} \cdot \left(t_H-d_i^{\mathrm{II}}\right), \forall i \tag{A15}$$

The relation between combinations of these two binary variables and the delivery date is demonstrated in Table A1.

The final selling price for batch i, PR_i is given in Eq. A16

$$PR_i=c_i^P \cdot \left(1-\frac{DT_i^{\mathrm{II}}}{d_i^{\mathrm{II}}-d_i^{\mathrm{I}}}\right) \tag{A16}$$

4. Sales and production cost

Sales can be obtained by summing up all the batch prices, as shown in Eq. A17

$$Sales=\sum_i PR_i \tag{A17}$$

Constraint (A18) shows the total production cost, which contains the cost for each batch. The cost of each individual batch i can be divided into two part, the fixed cost c_i^{Fix} and the variable cost C_i^{Var}. The sum of the fixed costs for all batches is denoted by c^{Fix}

$$Cost=\sum_i \left(c_i^{\mathrm{Fix}}+C_i^{\mathrm{Var}}\right)=c^{\mathrm{Fix}}+\sum_i C_i^{\mathrm{Var}} \tag{A18}$$

Table A1. Relation Between Combinations of Binary Variables and the Delivery Date

OW_i^{I}	OW_i^{II}	ODT_i
0	0	$0 \leq ODT_i \leq d_i^{\mathrm{I}}$
1	0	$d_i^{\mathrm{I}} \leq ODT_i \leq d_i^{\mathrm{II}}$
1	1	$d_i^{\mathrm{II}} \leq ODT_i \leq t_H$

Constraint (A19) denote that the variable cost for an individual batch i is the sum of the costs for all stages for this batch

$$C_i^{\mathrm{Var}}=\sum_{l \in L_i} PC_{il}, \forall i \tag{A19}$$

5. Objective function

In Constraint (A20), the objective is to maximize the profit which is the sales revenue minus the total cost

$$Profit = Sales-Cost \tag{A20}$$

Appendix B: Multistage Dynamic Optimization

In the sequential batch process, a product is manufactured by following its batch recipes, which consists of several operational stages. In the continuous process, the dynamic processes have fixed initial conditions and final values. In contrast, in the batch process, only the first stage in one batch recipe has the fixed initial conditions and only the last stage in the batch recipe has the fixed final values for its state variables. The initial conditions of a dynamic stage in the middle of a routing pathway are dependent on the final values of the previous operational dynamic stage.

A general multistage dynamic model for the sequential batch process can be described by a set of differential algebraic equations and various constraints.

Constraint (B1) is used to confine the time variable T_{il}, which starts from zero. τ_{il} denotes the end time of the stage l of batch i

$$0 \leq T_{il} \leq \tau_{il}, \quad \forall i, l \in L_i \tag{B1}$$

Equation B2 is a compact form of the differential equations, where X_{il} are state variables and U_{il} are input variables

$$\frac{dX_{il}(T_{il})}{dT_{il}}=f_{il}(X_{il}(T_{il}), U_{il}(T_{il})), \quad \forall i, l \in L_i \tag{B2}$$

Constraint (B4) is the path constraints for the state variables X_{il} and the input variables U_{il}. Path constraints usually consist of a set of inequalities for processing safety or the utility usage

$$h_{il}(X_{il}(T_{il}), U_{il}(T_{il})) \leq 0, \quad \forall i, l \in L_i \tag{B3}$$

Equation B4 shows the output equation. In batch process, the outputs are usually the concentrations of material components

$$Y_{il}(T_{il})=g_{il}(X_{il}(T_{il}), U_{il}(T_{il})), \quad \forall i, l \in L_i \tag{B4}$$

The criterion function (B5) is used to determine the processing cost φ_{il} of the dynamic process for stage l of batch i

$$\varphi_{il}=\phi_{il}(Y_{il}(T_{il})) \tag{B5}$$

Equation B6 is used to define the initial condition of each dynamic model. The initial points for the first dynamic stage for batch i is set as fixed parameters. The initial condition of other dynamic stage is set as the final value of the dynamic model in the previous stage

$$X_{il}(0)=\begin{cases} X_{i(l-1)}\left(\tau_{i(l-1)}\right), & l \geq 2 \\ x_i^0, & l=1 \end{cases}, \quad \forall i, l \in L_i \tag{B6}$$

Equation B7 is to define the final value in the final stage, where X_i^f denotes the product quality

$$X_i^f = X_{il}(\tau_{il}), \quad l = |L_i|, \ \forall i \tag{B7}$$

Constraint (B8) is the general quality requirement for the final value of the state variables

$$q_i\left(X_i^f\right) \le 0, \quad \forall i \tag{B8}$$

The criterion function is usually related with the entire time function. For example, in this work, the integral of the input with respect to time is used when we need the accumulated input value in the entire time interval. The final value for the criterion function is shown in Eq. B9

$$\varphi_{il} = \int_0^{\tau_{il}} U_{il}(T_{il}) dT_{il} \tag{B9}$$

Appendix C: Input Data for the Case Study

Models of reaction kinetics for Task 2 of Batch 1 is shown as follows.

$$M_{11} + M_{12} \xrightarrow{k_{12}^a} I_1 \tag{C1}$$

$$M_{11} + I_1 \xrightarrow{k_{12}^b} U_1 \tag{C2}$$

$$k_{12}^a = Z_{12}^a e^{-E_{12}^a/T_R} \tag{C3}$$

$$k_{12}^b = Z_{12}^b e^{-E_{12}^b/T_R} \tag{C4}$$

Table C1. Data of Scheduling Problem

Symbol	Description	Value	Unit
C_1^P	Price of Product 1	3.5 E + 03	\$/order
C_2^P	Price of Product 2	3.0 E + 03	\$/order
C_3^P	Price of Product 3	1.0 E + 03	\$/order
C_1^F	Fixed cost of Product 1	1.0 E + 03	\$/order
C_2^F	Fixed cost of Product 2	1.0 E + 03	\$/order
C_3^F	Fixed cost of Product 3	5.0 E + 02	\$/order
$C^{\rm HW}$	Cost of heating water	20	\$/m^3
$C^{\rm CW}$	Cost of heating water	2	\$/m^3
$D_1^{\rm I}$	First due date value of Product 1	6	h
$D_1^{\rm II}$	Second due date value of Product 1	9	h
$D_2^{\rm I}$	First due date value of Product 2	6	h
$D_2^{\rm II}$	Second due date value of Product 2	9	h
$D_3^{\rm I}$	First due date value of Product 3	6	h
$D_3^{\rm II}$	Second due date value of Product 3	9	h
O_1	Order demand of Product 1	2	Batch
O_2	Order demand of Product 2	2	Batch
O_3	Order demand of Product 3	2	Batch

Table C2. Fixed-Processing Time (h)

	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
Batch 1, Batch 2	1.0	NA	0.7	NA	1.0
Batch 3, Batch 4	NA	0.9	NA	1.5	NA
Batch 5, Batch 6	NA	1.2	NA	NA	NA

Table C3. Setup Time (h)

	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
Batch 1, Batch 2	0.1	0.1	0.1	0.1	0.1
Batch 3, Batch 4	0.1	0.1	0.1	0.1	NA
Batch 5, Batch 6	0.1	0.1	NA	NA	NA

Table C4. Parameters of Reaction Tasks

Task	Symbol	Description	Value	Unit
T_{12}	Z_{12}^a	Frequency factor	4E + 03	m^3/(mol h)
	Z_{12}^b	Frequency factor	1E + 08	m^3/(mol h)
	E_{12}^a	Normalized activation energy	5E + 03	K
	E_{12}^b	Normalized activation energy	1E + 04	K
	ΔH_{12}^a	Heat of reaction	230	kJ/mol
	ΔH_{12}^b	Heat of reaction	220	kJ/mol
	ρ_R^{12}	Density of reactor	8E + 02	kg/m^3
	$C_{P_R}^{12}$	Heat capacity of reactor	3	kJ/(kg K)
T_{13}	Z_{13}	Frequency factor	1E + 04	m^3/(mol h)
	E_{13}	Normalized activation energy	3E + 03	K
	ΔH_{13}	Heat of reaction	0	kJ/mol
	ρ_R^{13}	Density of reactor	8E + 02	kg/m^3
	$C_{P_R}^{13}$	Heat capacity of reactor	3	kJ/(kg K)
T_{21}	Z_{21}^a	Frequency factor	1E + 03	m^3/(mol h)
	Z_{21}^b	Frequency factor	5E + 02	m^3/(mol h)
	Z_{21}^c	Frequency factor	2E + 04	m^3/(mol h)
	E_{21}^a	Normalized activation energy	2E + 03	K
	E_{21}^b	Normalized activation energy	3E + 03	K
	E_{21}^c	Normalized activation energy	4E + 03	K
	ΔH_{21}^a	Heat of reaction	220	kJ/mol
	ΔH_{21}^b	Heat of reaction	0	kJ/mol
	ΔH_{21}^c	Heat of reaction	210	kJ/mol
	ρ_R^{21}	Density of reactor	1E + 03	kg/m^3
	$C_{P_R}^{21}$	Heat capacity of reactor	3.5	kJ/(kg K)
T_{22}	Z_{22}	Frequency factor	5E + 02	m^3/(mol h)
	E_{22}	Normalized activation energy	2E + 03	K
	ΔH_{22}	Heat of reaction	0	kJ/mol
	ρ_R^{22}	Density of reactor	1E + 03	kg/m^3
	$C_{P_R}^{22}$	Heat capacity of reactor	3.5	kJ/(kg K)
T_{31}	Z_{31}	Frequency factor	1E + 02	m^3/(mol h)
	E_{31}	Normalized activation energy	1E + 03	K
	ΔH_{31}	Heat of reaction	10	kJ/mol
	ρ_R^{31}	Density of reactor	1E + 03	kg/m^3
	$C_{P_R}^{31}$	Heat capacity of reactor	4	kJ/(kg K)

Table C5. Parameters of Reactors

Reactor	Symbol	Description	Value	Unit
$R_{\rm I}$	$V_R^{\rm I}$	Volume of reactor	5	m^3
	$V_J^{\rm I}$	Volume of jacket	1	m^3
	$\rho_J^{\rm I}$	Density of jacket	1E + 03	kg/m^3
	$C_{P_J}^{\rm I}$	Heat capacity of jacket	4.186	kJ/(kg K)
	$UA^{\rm I}$	Heat-transfer coefficient	8E + 04	kJ/(h K)
	$T_H^{\rm I}$	Temperature of heating water	370	K
	$T_C^{\rm I}$	Temperature of cooling water	300	K
	$T_{R,\max}^{\rm I}$	Maximum temperature of reactor	350	K
	$T_{J,\max}^{\rm I}$	Maximum temperature of jacket	370	K
	$F_{\max}^{\rm I}$	Maximum flow rate of heating and cooling water	20	m^3/h
$R_{\rm II}$	$V_R^{\rm II}$	Volume of reactor	5	m^3
	$V_J^{\rm II}$	Volume of jacket	1.5	m^3
	$\rho_J^{\rm II}$	Density of jacket	1E + 03	kg/m^3
	$C_{P_J}^{\rm II}$	Heat capacity of jacket	4.186	kJ/(kg K)
	$UA^{\rm II}$	Heat-transfer coefficient	1E + 05	kJ/(h K)
	$T_H^{\rm II}$	Temperature of heating water	370	K
	$T_C^{\rm II}$	Temperature of cooling water	300	K
	$T_{R,\max}^{\rm II}$	Maximum temperature of reactor	360	K
	$T_{J,\max}^{\rm II}$	Maximum temperature of jacket	370	K
	$F_{\max}^{\rm II}$	Maximum flow rate of heating water	30	m^3/h

Note that the parameter values of reactor $R_{\rm III}$ are identical to those of reactor $R_{\rm II}$.

Table C6. Data of Materials and Products

Production Line	Symbol	Material	Initial Material Concentration	Product	Symbol	Required Product Concentration
1	$C_{M_{11}}$	M_{11}	5E3 mol/m^3	P_1	C_{P_1}	4E3 mol/m^3
	$C_{M_{12}}$	M_{12}	5E3 mol/m^3			
2	C_{M_2}	M_2	2E3 mol/m^3	P_2	C_{P_2}	1.5 E3 mol/m^3
3	C_{M_3}	M_3	1E3 mol/m^3	P_3	C_{P_3}	0.9E3 mol/m^3

$$\frac{dC_{M_{11}}(t)}{dt}=-k_{12}^{a}C_{M_{11}}C_{M_{12}}-k_{12}^{b}C_{M_{11}}C_{I_1} \tag{C5}$$

$$\frac{dC_{M_{12}}(t)}{dt}=-k_{12}^{a}C_{M_{11}}C_{M_{12}} \tag{C6}$$

$$\frac{dC_{I_1}(t)}{dt}=k_{12}^{a}C_{M_{11}}C_{M_{12}}-k_{12}^{b}C_{M_{11}}C_{I_1} \tag{C7}$$

$$\frac{dC_{U_1}(t)}{dt}=k_{12}^{b}C_{M_{11}}C_{I_1} \tag{C8}$$

Models of reaction kinetics for Task 4 of Batch 1 is shown as follows.

$$I_1\xrightarrow{k_{14}}P_1 \tag{C9}$$

$$k_{14}=Z_{14}e^{-E_{14}/T_R} \tag{C10}$$

$$\frac{dC_{I_1}(t)}{dt}=-k_{14}C_{I_1} \tag{C11}$$

$$\frac{dC_{P_1}(t)}{dt}=k_{14}C_{I_1} \tag{C12}$$

Models of reaction kinetics for Task 1 of Batch 2 is shown as follows.

$$M_2\underset{k_{21}^{b}}{\overset{k_{21}^{a}}{\rightleftharpoons}}I_2 \tag{C13}$$

$$I_2\xrightarrow{k_{21}^{c}}U_2 \tag{C14}$$

$$k_{21}^{a}=Z_{21}^{a}e^{-E_{21}^{a}/T_R} \tag{C15}$$

$$k_{21}^{b}=Z_{21}^{b}e^{-E_{21}^{b}/T_R} \tag{C16}$$

$$k_{21}^{c}=Z_{21}^{c}e^{-E_{21}^{c}/T_R} \tag{C17}$$

$$\frac{dC_{M_2}(t)}{dt}=-k_{21}^{a}C_{M_2}+k_{21}^{b}C_{I_2} \tag{C18}$$

Table C7. Temperature Specification

Production Line	Initial Temperature (K)	Final Temperature (K)
1	300	320
2	300	320
3	300	320

$$\frac{dC_{I_2}(t)}{dt}=k_{21}^{a}C_{M_2}-\left(k_{21}^{b}+k_{21}^{c}\right)C_{I_2} \tag{C19}$$

$$\frac{dC_{U_2}(t)}{dt}=k_{21}^{c}C_{I_2} \tag{C20}$$

Models of reaction kinetics for Task 3 of Batch 2 is shown as follows.

$$I_2\xrightarrow{k_{22}}P_2 \tag{C21}$$

$$k_{23}=Z_{23}e^{-E_{23}/T_R} \tag{C22}$$

$$\frac{dC_{I_2}(t)}{dt}=-k_{23}C_{I_2} \tag{C23}$$

$$\frac{dC_{P_2}(t)}{dt}=k_{23}C_{I_2} \tag{C24}$$

Models of reaction kinetics for Task 1 of Batch 3 is shown as follows.

$$M_3\xrightarrow{k_{31}}P_3 \tag{C25}$$

$$k_{31}=Z_{31}e^{-E_{31}/T_R} \tag{C26}$$

$$\frac{dC_{M_3}(t)}{dt}=-k_{31}C_{M_3} \tag{C27}$$

$$\frac{dC_{P_3}(t)}{dt}=k_{31}C_{M_3} \tag{C28}$$

Manuscript received Apr. 16, 2015, and revision received July 6, 2015.