Integrated Scheduling and Dynamic Optimization of Batch Processes Using State Equipment Networks

Yisu Nie and Lorenz T. Biegler
Dept. of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

John M. Wassick
The Dow Chemical Company, Midland, MI 48674

DOI 10.1002/aic.13738
Published online February 14, 2012 in Wiley Online Library (wileyonlinelibrary.com).

A systematic framework for the integration of short-term scheduling and dynamic optimization (DO) of batch processes is described. The state equipment network (SEN) is used to represent a process system, where it decomposes the process into two basic kinds of entities: process materials and process units. Mathematical modeling based on the SEN framework invokes both logical disjunctions and operational dynamics; thus the integrated formulation leads to a mixed-logic dynamic optimization (MLDO) problem. The integrated approach seeks to benefit the overall process performance by incorporating process dynamics into scheduling considerations. The solution procedure of an MLDO problem is also addressed in this article, where MLDO problems are translated into mixed-integer nonlinear programs using the Big M reformulation and the simultaneous collocation method. Finally, through two case studies, we show advantages of the integrated approach over the conventional recipe-based scheduling method. © 2012 American Institute of Chemical Engineers *AIChE J,* 58: 3416–3432, 2012
Keywords: batch processes; scheduling; dynamic optimization; MINLP; collocation

Introduction

Batch processes have received considerable attention over the past few decades. Inherent flexibility enables batch processing to be widely applied to the manufacture of a great variety of chemical and pharmaceutical products. Traditionally, scheduling and dynamic optimization (DO) of a batch process are approached independently. In the scheduling phase, a production schedule is designed on the basis of fixed recipes, in which operating policies for all processing stages of a batch process are stipulated, such as processing times and control inputs. In this approach, called recipe-based scheduling, batch operations are permissible to be modeled by linear equations in terms of material balance, and their dynamic nature has been circumvented. With a few exceptions (e.g., Chachuat et al.[1]), dynamic optimization of batch operations has been investigated only within a single unit environment, in the absence of complete manufacturing sequences. However, the rapidly changing global markets and intensive competitive environment urge drastic modifications in current production strategies. Integrated decision-making approach is currently playing a crucial role for many chemical companies for the sake of flexibility and information availability at all levels of production.[2]

A vast body of the aforementioned recipe-based scheduling problems has been considered over the past two decades. For a batch process to be scheduled, its flow sheet is usually represented by networked structures utilizing bipartite graphs, e.g., two prevalent representation schemes called the state task network (STN) proposed by Kondili et al.[3] and resource task network (RTN) proposed by Pantelides.[4] Optimization formulations based on these structures are mostly posed as mixed integer linear programs (MILPs),[5–8] or in certain circumstances, with a few nonlinearities.[9,10] The discrete variables primarily account for tactical decisions such as unit-task assignment and task sequencing. With the fruitful development of formulation techniques and state-of-the-art MILP solvers, a practitioner is able to handle fairly large instances within reasonable computational times. However, recipe-based scheduling methods pertain to suboptimal and unreliable production schedules for real-world problems due to the rigid operating strategy. A batch production sequence is not completely optimized at the operation level in this approach, because important degrees of freedom have been discarded from the process. Meanwhile, a schedule remains feasible only if every comprised processing stage has been executed precisely according to its recipe, which, often is unrealizable in the face of process disturbances and uncertainties.[11] As a matter of fact, a trivial variation or delay of an operation may incur sharp profit losses or even failure of an manufacturing facility, when no adjustments of downstream operations are allowed.

A foreseeable solution to mitigate the problem is to recover the dynamic characteristics of batch operations, and directly incorporate them into scheduling considerations. Bhatia and Biegler[12] proposed an integrated formulation for

Correspondence concerning this article should be addressed to L. T. Biegler at biegler@cmu.edu.

the simultaneous design, scheduling and optimal control of multiproduct batch plants. In this work, the scheduling problem was discussed in the scope of flowshop plants and a special class of transfer policies, and fixed batch recipes were replaced with detailed dynamic models, which contributed to the improvement of profitability at a remarkable level. A more recent work was presented by Mishra et al.,[13] where they compared the so-called standard recipe approach (SRA) with the overall optimization approach (OOA) in optimizing general batch plants. The scheduling model was formulated per the unit-specific continuous-time STN representation,[14] where different kinds of transportation and storage policies can be handled. They fitted dynamic models of the reaction tasks into the STN framework in the OOA, discretized the resulting mixed-integer dynamic optimization (MIDO) problem using a finite difference method and solved it as a mixed-integer nonlinear program (MINLP). They tested both the SRA and the OOA in two case studies. In the results, the OOA achieved better solutions than the SRA. However, as the authors also highlighted, computational load became the bottleneck of the OOA that prohibited its practical use.

The objective of this article is to propose a systematic and computationally efficient approach that applies to the integration of scheduling and dynamic optimization of batch processes. To start, we first elucidate how both facets can be tied together in the context of a state equipment network (SEN), which is a flow sheet superstructure we rely on to account for the graphical and also mathematical representation of a process. In addition, we show the development of a hybrid discrete-continuous model based on the SEN that eventually gives rise to a mixed-logic dynamic optimization problem (MLDO).[15] Solution methods of MLDOs are discussed along with reformulation techniques for disjunctions and differential algebraic equations (DAEs). Many successful research studies on discrete dynamic systems have been reported from various fields, e.g., integrated design and control,[16] synthesis of rice drying processes[17] and simultaneous scheduling and control optimization in polymer grade transitions.[18,19] In this work, we elaborate on a procedure to translate MLDOs to MINLPs, which is coupled with the Big M reformulation approach[20] and the simultaneous collocation strategy.[21] Thus, the problem can be solved with the help of off-the-shelf MINLP solvers. Finally, two illustrative examples are studied to demonstrate the strengths of the integrated approach.

Problem Statement

The integrated approach aims at improving the overall performance of a batch process by simultaneously optimizing its production schedule and corresponding control profiles of the process units involved. The problem can be stated as follows:

Given:

Plant configuration

- a set of process units and their usages
- feed materials, intermediate and final products and their constituents
- information on production sequences or steps

Scheduling preknowledge

- a fixed scheduling time horizon (except for makespan problems)
- target performance index, e.g., maximum profitability, Minimum makespan or earliness
- transfer policy for each material
- procurement price of raw materials
- market price, demand and quality requirements of final products
- market price of utilities consumed in batch operations

Process dynamics

- dynamic (or simplified) models of process operations
- bounds on state and control variables corresponding to possible process operations
- additional constraints regarding safety and quality issues

Determine:

- optimal production sequences and timings of all batch units
- optimal control profiles of batch units
- optimal performance index value

Admittedly, a desired comprehensive scheduling formulation is expected to handle many aspects of manufacturing considerations: order fulfillment, product transitions and shared resources, to name a few. Albeit simplified, we will restrict our problem to a more concise status, where the objective is to maximize the net profit at the end of scheduling horizon with no change-over delays and fulfillment deadlines, while the unlimited intermediate storage (UIS) policy is assumed. Nonetheless, extensions can be carried out within the SEN infrastructure, using similar techniques developed for the STN and the RTN.[22]

The state equipment network

Conventionally, short-term chemical batch process scheduling problems have been represented via network structures using the ‘‘task’’ as one of their basic elements. The term task refers to any abstracted process operation that is able to change chemical or physical properties of process materials for manufacturing purposes, e.g., heating, reaction and separation. Production schedules are designed by allocating all possible tasks over the time domain under the guidance of product recipes and scheduling algorithms. The rationale for introducing the task concept is to avoid the inclusion of complex dynamic details of equipment units and their operations, and therefore one can easily focus at the tactical level decisions of a process. But in the integrated scheme, process equipment becomes a better reference for the description of process dynamics than processing tasks. Because in this circumstance of existing equipment, it is conducive to specify the configuration of equipment first, and leave the selection of process operations for each unit as a second stage decision. And also, process variables are directly associated with concrete process units rather than conceptually built processing tasks; mathematically, all state and control variables in dynamic models persist as physically meaningful entries only when they are defined with respect to batch units. Thus, to open the door to integration, an alternative equipment-oriented perspective is needed. The state equipment network framework suits the context. In the SEN, complete connectivity of all process units is postulated and these units are described in the form of rigorous models without preassigned tasks.[23] The SEN was originally intended for process design and synthesis problems, and quickly found its way into a wide array of applications, particularly in continuous chemical processes.[24,25] One of its major advantages is that it provides a consistent modeling strategy and proper handling of units that operate under different operating modes.[26]

The original SEN framework adopts a directed graph with two basic types of nodes to represent a batch process. These

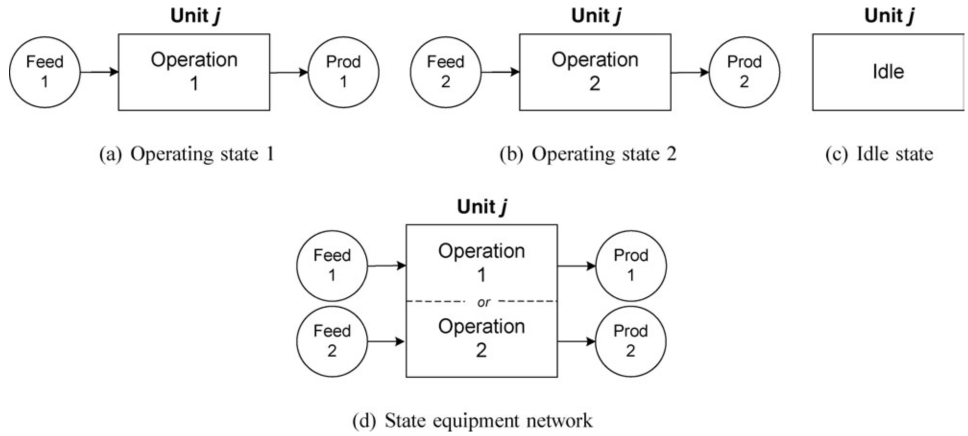


Figure 1. An illustrative example of applying the state equipment network.

nodes are namely, the state nodes, including all sorts of materials involved in the process; and the equipment nodes, containing all processing units in the plant. However, a striking feature of batch plants is their capability of flexible manufacturing, which invokes discrete time-varying behavior of process units, such as switching between different operations, start-ups and shut-downs. As a result, the mathematical description of an equipment node is subject to change over time as well as the connectivity of a SEN graph. Therefore, an additional temporal dimension of combinatorial decisions has to be incorporated into the original SEN to designate the status of equipment nodes. As equipment and material state nodes are connected by directed edges alternatively, a scheduling problem can be reinterpreted as routing material flows through the network. Last but not least, since a batch unit may accommodate multiple operations, the term state here can also be defined as an operational status of the unit. To avoid ambiguity, we will use the term operating state to indicate operation and the term material to refer to material state.

To better interpret how the SEN tackles a batch scheduling problem, a simple instance of a single-stage batch plant is depicted in Figure 1. The unit serves for two different manufacturing purposes. It converts the corresponding feed material to the product, if switched to a particular operating state. This plant has three possible setups, shown in Figures 1a-c. The SEN representation drawn in Figure 1d is able to capture all these three potential configurations of the system and postulate them together with disjunctive propositions (the idle state is included implicitly).

By definition, the SEN disaggregates a process flow sheet into materials and equipment with time variant operating modes to attack batch scheduling problems. Therefore, designing a schedule can be also conceived as managing the sequence of equipment operating states along with material distribution. As a result, a production schedule declares when and how to switch unit operating states while material transportation is executed instantaneously as an accompaniment of state transitions. In certain occasions where the executing times for material transfers cannot be neglected, the transfer operations can be treated explicitly as unit operating states.

Model formulation

The discussion of the integrated formulation is staged into two steps. First, starting with considering the problem as a whole, we show how the scheduling approach use the SEN to sequence, size and time batch operations. Most of the efforts are devoted to issues like material balance in network, assignment of operating states and measurement of material qualities. These site-wide level decisions are essential to line up a manufacturing sequence. In the next section, dynamic models of operating states are explored to disclose further details in the operation phase, and dynamic optimization of unit operations is performed with these models.

Scheduling using the SEN

Time representation, as a key criterion to distinguish scheduling formulations, is open to many choices.[27] It determines the skeleton of the resulting optimization models. To better suit the equipment control perspective, the model in this article adopts the unit-specific event-based continuous-time representation,[28] where a scheduling horizon is divided up into a finite number of event slots for each unit. These slots are named in the sense that they have the capability to turn operating states on and off at the beginning and the end of them, respectively. These slots can be asynchronous from one unit to another, thus the scheduling formulation asks for a group of sequencing constraints to adjust the global sequencing behavior among different pieces of equipment. Moreover, for each process unit, its operating states in event slots are labeled with a series of binary variables $w_{j,s,n}$. The remainder of this section will take the reader through the five major aspects of scheduling considerations in the SEN.

Assignment Constraints. In these constraints, operating states are allocated in both units and time horizon.

$$\sum_{s\in\mathcal{S}_j} w_{j,s,n} \leq 1 \quad \forall j\in\mathcal{J}, n\in\mathcal{N} \tag{1}$$

In Eq. 1, binary variable $w_{j,s,n}$ indicates unit j is in operating state s in current time slot n, if equal to 1. In the SEN representation, operating states of the same unit are exclusive of each other because equipment can not be shared. In addition, it is also possible that the unit is not occupied by any operating states during certain periods of time such that all binaries are equal to 0. In short, the assignment of operating states allows at most one operating state to be active in a unit per event slot.

Material Balance. An event point is defined as the time point where an event slot starts. At such a point, balance equations are enforced for every material r to keep track of its available amount, before moving into the next time interval. Moreover, at event point n, materials that are not used by any process operation executed in event slot n are termed as excess materials $E_{r,n}$.

$$E_{r,n} = E_{r,n-1} + \sum_{j \in \mathcal{J}_r^{\mathrm{p}}} R^p_{j,r,n-1} - \sum_{j \in \mathcal{J}_r^{c}} R^c_{j,r,n} \quad \forall r \in \mathcal{R}, n \in \mathcal{N}, n > 1 \tag{2a}$$

$$E_{r,1} = E_{r,0} - \sum_{j \in \mathcal{J}_r^{\mathrm{c}}} R^{\mathrm{c}}_{j,r,1} \qquad \forall r \in \mathcal{R} \tag{2b}$$

$$E_{r,0} = \overline{E}_r \qquad \forall r \in \mathcal{R}' \subseteq \mathcal{R} \tag{2c}$$

Equation 2a states that the amount of excess material r at event point n equals its value at the previous event point $n - 1$ adjusted by the amount consumed within slot n, and produced within slot $n - 1$; however, at the first event point, no production is yet to be obtained (Eq. 2b). A process material can be generated or consumed by one or several units, denoted as subsets $\mathcal{J}_r^p$ and $\mathcal{J}_r^c$ respectively. The initial amount of material r ($E_{r,0}$) may be specified accordingly as stated in Eq. 2c. Usually, the initial amounts of intermediate and final materials are assigned to the given parameters $\bar{E}_r$, while the amounts of raw materials of procurement cost remain variables, i.e., in Eq. 2c set $\mathcal{R}' = \mathcal{R} \backslash \mathcal{R}_{\mathrm{raw}}$.

Capacity Constraints. Typically, the extent of a batch operation is limited by the vessel size of the equipment used and also pertinent process safety considerations. Equation 3 helps enforce these restrictions on the batch size $b_{j,n}$.

$$\sum_{s \in \mathcal{S}_j} w_{j,s,n} B_j^{\min} \le b_{j,n} \le \sum_{s \in \mathcal{S}_j} w_{j,s,n} B_j^{\max} \qquad \forall j \in \mathcal{J}, n \in \mathcal{N} \tag{3}$$

Meanwhile, the amount of excess material E_r must stay within a certain range that is stipulated by regulations such as safety stock, storage limit and so forth (Eq. 4). Like other unit-specific formulations, E_r does not necessarily agree with the material inventory in real time, since event points of units are most likely asynchronized. In this formulation, we avoid this problem by assuming the UIS policy. Nevertheless, rigorous treatment of other operational policies can be amended at the cost of additional variables and constraints.[29]

$$E_r^{\min} \le E_{r,n} \le E_r^{\max} \qquad \forall r \in \mathcal{R}, n \in \mathcal{N} \tag{4}$$

Timing Constraints. In this continuous-time formulation, the nth event point of batch unit j is denoted as $T_{j,n}$ and the associated processing time of the unit is called $Tp_{j,n}$. Since the number of these timing variables is proportional to the number of batch units, the SEN generally leads to smaller size models compared to the conventional task-oriented formulations. To sequence operations, Eq. 5 is first applied to individual units. Here, no overlapping is allowed for any neighboring pair of event slots of a unit, i.e., the starting time of an event slot cannot be earlier than the time when its precedent slot ends.

$$T_{j,n+1} \ge T_{j,n} + Tp_{j,n} \qquad \forall j \in \mathcal{J}, n \in \mathcal{N}, n < N \tag{5}$$

Besides, to obtain appropriate alignments of event slots of different units that use the same intermediate material, a second group of constraints is introduced. For instance, if material r is produced by unit j at the end of time slot n, and another unit j' has active operating state s' in event slot n' ($n < n'$) that consumes the material, then event point n' of unit j' should be placed after the end of slot n of unit j in real time.

$$T_{j',n'} \ge T_{j,n} + Tp_{j,n} - H\left(2 - \sum_{s \in \mathcal{S}_j \cap \mathcal{S}_r^{\mathrm{p}}} w_{j,s,n} - \sum_{s' \in \mathcal{S}_{j'} \cap \mathcal{S}_r^{\mathrm{c}}} w_{j',s',n'}\right)$$
$$\forall r \in \mathcal{R}, j \in \mathcal{J}_r^{\mathrm{p}}, j' \in \mathcal{J}_r^{\mathrm{c}}, j \ne j', n, n' \in \mathcal{N}, n < n' \le N \tag{6}$$

This constraint is only enforced when both the production and consumption actions take place, thus it is relaxed by a big M term with constant H, which is the given scheduling horizon. And also, all processing operations should be completed within the specified time horizon for a valid schedule. For this reason, a group of bounding constraints are introduced as Eqs. 7 and 8.

$$T_{j,n} \le H \qquad \forall j \in \mathcal{J}, n \in \mathcal{N} \tag{7}$$

$$T_{j,N} + Tp_{j,N} \le H \qquad \forall j \in \mathcal{J} \tag{8}$$

Quality Measurement. This set of variables and constraints does not appear in conventional batch scheduling models, due to the assumption of well-executed recipes. Under this assumption, no quality giveaways will occur after running a batch for multiple times, and therefore there is no need to retain quality information. However, when operational strategy varies in time, it consequently results in variations in material quality, such as concentration distribution in a mixture. To account for this in the integrated scheme, we define $\eta_{r,c,n}$ as the fraction of component c in excess compound material r at event point n, as well as its counterpart $\phi_{j,r,c,n}$ for material r produced by unit j at the end of event slot n. Material flows coming from different sources are likely to have nonidentical composition distributions. A way to deal with this is to assume a blending before using regulation, that is, newly produced materials are fully blended with inventories before they are used in the next processing stage. Under this circumstance, an excess mixture after blending has the averaged composition of all source flows, and the quality of the materials is measured by the mean value as shown in Eq. 9.

$$\eta_{r,c,n} = \frac{\left(\eta_{r,c,n-1} E_{r,n-1} + \sum_{j \in \mathcal{J}_r^{\mathrm{p}}} \phi_{j,r,c,n-1} R^{\mathrm{p}}_{j,r,n-1}\right)}{\left(E_{r,n-1} + \sum_{j \in \mathcal{J}_r^{\mathrm{p}}} R^{\mathrm{p}}_{j,r,n-1}\right)}$$
$$\forall r \in \mathcal{R}, c \in \mathcal{C}_r, n \in \mathcal{N}, n > 1 \tag{9}$$

On the other hand, to ensure specific requirements of material quality, especially for final products, additional constraints are enforced to restrict $\eta_{r,c,n}$. These constraints can be generally written as Eq. 10.

$$H(\eta_{r,c,n}, \overline{\eta}_{r,c}) \ge 0 \qquad \forall r \in \mathcal{R}, c \in \mathcal{C}_r, n \in \mathcal{N} \tag{10}$$

Without a doubt, many other quality targets can be applied, such as physical and mechanical properties. This is permissible through the use of the SEN, because the quality of a (final or intermediate) product can be essentially related back to the operating condition of associated operating states. Mathematically, the quality can be written as a function of the state and control variables of a dynamic process model, which we discuss in the following section.

DO in the SEN

Dynamic optimization for a given batch operation batch operation determines its optimal control profile in continuous time. To date, numerous rigorous optimization applications have been successfully practiced in continuous process industries. However the extension to batch processes encounters great challenges, due in part to the discrete nature of batch operations. The SEN-based framework offers a great opportunity to model such a hybrid discrete-continuous system. For an operation, the dynamic model is couched in continuous time with a finite time length. In terms of process scheduling, the discrete decisions are represented via discrete events taking place at distinct time points. Logical disjunctions are applied to tie the discrete and the continuous parts together. On the other hand, it is also worth to note that dynamic models at different knowledge levels of process details are generally applicable in the SEN, from the most sophisticated first-principle dynamic models[30] to reduced order models[31] of various degrees of accuracy. An appropriate choice among these models is also a critical issue to balance the information coverage and the computational complexity of an integrated model. In this article, we use first-principle dynamic models, by virtue of their accuracy over wide operating ranges.

Dynamic Models of Operating States. The residence time of a batch operation is finite but unknown in advance, which means, DO of an active operation state is performed within a receding time horizon of a variable length. Transforming the time coordinate into unit length includes normalizing the integration time to $\tau \in [0, 1]$ and applying the chain rule to all time derivatives. With this, we can further write the dynamic model of a operating state as a generic DAE system with bounded states, controls and processing times.

$$\left.\begin{aligned} \frac{dz_{j,n}(\tau)}{d\tau} &= f_{j,s}(z_{j,n}(\tau), y_{j,n}(\tau), u_{j,n}(\tau))Tp_{j,n} \\ g_{j,s}(z_{j,n}(\tau), y_{j,n}(\tau), u_{j,n}(\tau)) &= 0 \\ z_{j,s}^{\min} \le z_{j,n}(\tau) &\le z_{j,s}^{\max} \\ y_{j,s}^{\min} \le y_{j,n}(\tau) &\le y_{j,s}^{\max} \\ u_{j,s}^{\min} \le u_{j,n}(\tau) &\le u_{j,s}^{\max} \\ Tp_{j,s}^{\min} \le Tp_{j,n}(\tau) &\le Tp_{j,s}^{\max} \end{aligned}\right\} \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, n \in \mathcal{N} \tag{11}$$

In Eq. 11, all the variables are defined over unit j. However for different operating states of the unit, the dynamic models are constructed in dissimilar manners. Furthermore, the dynamic DAEs used here are preferred to be of index one and reformulation is recommended for higher index systems, so as to guarantee solution uniqueness and numerical robustness.[32]

In addition to Eq. 11, the initial condition of the differential state variable $z_{j,n}$ must be specified. Both the batch size and the inlet material quality are considered as the input variables to the dynamic system, given in Eq. 12.

$$z_{j,n}(0) = Z_{j,s}(b_{j,n}, \eta_{r,c,n}) \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, r \in \mathcal{R}_s^{c}, c \in \mathcal{C}_r, n \in \mathcal{N} \tag{12}$$

Material production and consumption are important variables that link the intrinsic dynamic system of a batch unit to external plant environment. Often in practice, input materials are considered to be consumed at the beginning ($\tau = 0$) while output products become available when an operation ends ($\tau = 1$). And over the time interval, the batch unit is occupied. Here, material inputs and outputs are described via functions of state variables as shown in Eq. 13. However, for semi-batch operations where feed materials are continuously transferred into a unit, the consumption functions can be integrals over the residence time.

$$\begin{aligned} R_{j,r,n}^{p} &= R_{j,s}^{p}(z_{j,n}(1), y_{j,n}(1)) \quad && \forall j \in \mathcal{J}, s \in \mathcal{S}_j, r \in \mathcal{R}_s^{p}, n \in \mathcal{N} \\ R_{j,r,n}^{c} &= R_{j,s}^{c}(z_{j,n}(0), y_{j,n}(0)) \quad && \forall j \in \mathcal{J}, s \in \mathcal{S}_j, r \in \mathcal{R}_s^{c}, n \in \mathcal{N} \end{aligned} \tag{13}$$

Likewise, product quality can be written as functions of state variables as well. Moreover, a quality measurement function may be path dependent, i.e., in the form of an integral over time $\tau = [0, 1]$. In general, Eq. 14 is able to represent product quality with proper definitions of state variables.

$$\phi_{j,r,c,n} = \phi_{j,s}(z_{j,n}(\tau), y_{j,n}(\tau)) \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, r \in \mathcal{R}_s^{p}, c \in \mathcal{C}_r, n \in \mathcal{N}, \tau \in [0, 1] \tag{14}$$

Like excess materials, a product material of an operation may also be subject to certain quality standards, given by parameter $\overline{\phi}_{j,r,c}$. Usually, quality giveaways are allowed only within a tolerance and quality restrictions are enforced (Eq. 15).

$$\Phi_{j,s}(\phi_{j,r,c,n}, \overline{\phi}_{j,r,c}) \ge 0 \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, r \in \mathcal{R}_s^{p}, c \in \mathcal{C}_r, n \in \mathcal{N} \tag{15}$$

Taking the operating cost of an operation into account is also important for the integrated formulation, since it is one of the major concerns in running a batch plant. In particular, utility charges are a dominant fraction of the total cost, so that in this study we consider them as the main subject. In a deterministic scenario without price fluctuation, the cost of running unit j within slot n ($F_{j,n}$) depends on the operational conditions: the active operating state, process control inputs, execution times and batch size.

$$F_{j,n} = F_{j,s}(u_{j,n}(\tau), Tp_{j,n}, b_{j,n}) \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, n \in \mathcal{N}, \tau \in [0, 1] \tag{16}$$

Disjunctions of Batch Units. As we described earlier, in contrast to task-oriented networks, the SEN regards different operating states as individual dynamic systems inside disjuncts but with a consistent variable definition. Therefore, the disjunction for a unit comprises the disjuncts of the operating states that the unit can perform. Besides, one additional disjunct is added to account for equipment idling. In the idle state, no dynamic equations are needed to describe the behavior of the unit, and all of the interfacing variables such as material production and consumption are fixed to zero. The modeling approach is usually addressed as generalized disjunctive programming (GDP).[33] It recasts discrete optimization problems into logic-based models, offering structural advantages both in model formulation and solution.[34] In the context of the SEN, a disjunction captures the essence of exclusive choices between operating states with the assistance of assignment constraints. With this approach, dynamic models of operating states are organized by disjunctions as Eq. 17 states.

$$\cdots \left[\begin{array}{c} w_{j,s,n}=1 \\ \dfrac{dz_{j,n}(\tau)}{d\tau}=f_{j,s}(z_{j,n}(\tau),y_{j,n}(\tau),u_{j,n}(\tau))Tp_{j,n} \\ g_{j,s}(z_{j,n}(\tau),y_{j,n}(\tau),u_{j,n}(\tau))=0 \\ z_{j,s}^{\min}\le z_{j,n}(\tau)\le z_{j,s}^{\max} \\ y_{j,s}^{\min}\le y_{j,n}(\tau)\le y_{j,s}^{\max} \\ u_{j,s}^{\min}\le u_{j,n}(\tau)\le u_{j,s}^{\max} \\ Tp_{j,s}^{\min}\le Tp_{j,n}(\tau)\le Tp_{j,s}^{\max} \\ z_{j,n}(0)=Z_{j,s}(b_{j,n},\eta_{r,c,n}) \\ R_{j,r,n}^{\mathrm{p}}=R_{j,s}^{\mathrm{p}}(z_{j,n}(1),y_{j,n}(1)) \\ R_{j,r,n}^{\mathrm{c}}=R_{j,s}^{\mathrm{c}}(z_{j,n}(0),y_{j,n}(0)) \\ \phi_{j,r,c,n}=\phi_{j,s}(z_{j,n}(\tau),y_{j,n}(\tau)) \\ F_{j,n}=F_{j,s}(u_{j,n}(\tau),Tp_{j,n},b_{j,n}) \\ \Phi_{j,s}(\phi_{j,r,c,n},\phi_{j,r,c})\ge 0 \end{array}\right]_{s\in\mathcal{S}_j} \cdots \bigvee \left[\begin{array}{c} \sum_{s\in\mathcal{S}_j} w_{j,s,n}=0 \\ R_{j,r,n}^{\mathrm{p}}=0 \\ R_{j,r,n}^{\mathrm{c}}=0 \\ Tp_{j,n}=0 \\ F_{j,n}=0 \end{array}\right] \quad \forall j\in\mathcal{J},n\in\mathcal{N},\tau\in[0,1] \tag{17}$$

Integrated scheduling and DO

In the integrated formulation, the objective function is to maximize the revenue for a given scheduling horizon, which is equal to product sales minus raw materials and operating costs. The total amount of products that are made to sell are counted at the end of the last event slot. Meanwhile feedstock is purchased in advance and cost of unit operations is accumulated along the time horizon. On the other hand, the constraints of the integrated formulation come from both the scheduling and unit operation respects. In sum, the problem can be stated as Eq. 18.

$$\begin{array}{lll} \max & \sum_{r\in\mathcal{R}_{\text{prod}}} P_r(E_{r,N}+\sum_{j\in\mathcal{J}_r^{\mathrm{p}}} R_{j,r,N}^{\mathrm{p}})-\sum_{r\in\mathcal{R}_{\text{raw}}} P_rE_{r,0}-\sum_{j\in\mathcal{J},n\in\mathcal{N}} F_{j,n} & \\ & \text{Assignment constraints} & \text{Eq. 1} \\ & \text{Material balance} & \text{Eq. 2} \\ & \text{Capacity constraints} & \text{Eqs. 3,4} \\ s.t. & \text{Timing constraints} & \text{Eqs. 5}-8 \\ & \text{Material quality} & \text{Eqs. 9,10} \\ & \text{Unit operation} & \text{Eq. 17} \end{array} \tag{18}$$

Tightening Constraints. A good lower bound from solving relaxed problems is always preferable for solving a mixed-integer program. For this reason, a number of redundant constraints are recommended to be added to the foregoing formulation. These auxiliary constraints help improve the tightness of the formulation and therefore accelerate optimization search.

Tightening timing constraints. As studied in the context of conventional batch scheduling problems in linear form, timing constraints are a critical determinant of the tightness of the corresponding relaxed linear programs.[29] We keep the use of a good tightening constraint, which states that the summation of processing times of any unit over all event slots needs to be less or equal than the length of the scheduling horizon.

$$\sum_{n\in\mathcal{N}} Tp_{j,n}\le H \qquad \forall j\in\mathcal{J} \tag{19}$$

Mass balance of a unit. Although the production and consumption of materials are (generally nonlinear) functions of state variables, the linear mass balance relationship still holds for each process unit. The rationale for preserving the mass balance equations is to assure the material conservation law for the relaxed problem, where the integrality requirement on binary variables is dismissed. In Eq. 20, balance equations are written with respect to individual units.

$$\sum_{r\in\mathcal{R}_j^{\mathrm{p}}} R_{j,r,n}^{\mathrm{p}}=\sum_{r\in\mathcal{R}_j^{\mathrm{c}}} R_{j,r,n}^{\mathrm{c}} \qquad \forall j\in\mathcal{J},n\in\mathcal{N} \tag{20}$$

Overall Formulation. With all the elements at hand, the overall formulation is ready to be demonstrated. It presents as an MLDO problem, including Eqs. 18–20. The formulation carries out scheduling and DO simultaneously, and there are two important factors that need attention when applying this formulation. First, as pointed out in Li and Floudas,[35] a proper number of event points is conducive to balance between the optimality and the computational complexity of an event-based scheduling formulation. Second, the scale of dynamic models used in the formulation is of equal importance in determining its performance.

Reduced Models of Recipe-Based Operating States. For batch units that do not require dynamic optimization in the SEN, the recipe-oriented strategy can still be accommodated by reducing dynamic models. As the process control inputs and processing times are fixed according to recipes, state profiles are only functions of initial conditions of

Figure 2. Reformulation of MLDOs into MINLPs.

differential variables. Therefore for a dynamic model of an operating state, its performance indices such as production, consumption and operating cost can be reduced to functions of the batch size and quality variables. These functions are often in simple algebraic form. Furthermore, quality measurements of materials become constants and constraints on quality measurements can be simplified or even discarded, when all operation sequences are restricted by recipes. Under this circumstance, the batch size variables are the only ones left to quantify the performance of operating states, where linear equations are frequently encountered. This gives rise to a reduced problem without dynamic and quality variables. The reduced problem can be conceived as a typical recipe-based batch scheduling problem and recast as an MILP. It is also worth to note that any solution of the reduced problem completed with the constant values of the dynamic and quality variables is still a feasible solution of the corresponding full space problem. In fact, the best of these solutions is often a good initial guess to start with for solving the full problem.

Solution Strategy. The integrated model is in the MLDO form which requires reformulation so as to be solved by existing optimization algorithms and solvers. On one hand, one needs to cope with DAE systems; on the other, disjunctions also demand relevant treatments. Several approaches with complementary strengths are developed for these two purposes, as discussed below.

There exist two ways to reformulate logical disjunctions, namely, the Big M method and the convex hull relaxation (CHR). The convex hull relaxation generally leads to a tighter relaxed problem at cost of a larger problem size, compared with the Big M method. Hence, the performance of these two approaches is problem dependent. In the integrated SEN formulation, however, dynamic models of operating states are generally nonlinear and nonconvex, such that applying the CHR risks cutting off the feasible region of an MLDO. In addition, the Big M method can avoid disaggregating continuous variables that mainly consist of the states and controls of dynamic models. Furthermore, we choose an enhanced version of the Big M method called the Multi-M reformulation. In this approach, nonidentical *M*s are allowed to be associated with different equations in a disjunct, and they are set to values as small as possible, while being sufficiently large to preserve equivalent feasibility. This technique provides a tighter reformulation than the conventional Big M approach without increasing the size of a problem.[36]

An integrated scheduling problem is translated to MIDO form after transforming disjunctions. In practice, MIDOs can be solved either in a decomposed manner or in a discretized full space. Decomposition techniques, such as the generalized Benders decomposition (GBD) and the outer approximation (OA), split an MIDO instance into two parts: a master MILP problem and a primal dynamic optimization (PDO) problem. The dynamic subproblem is usually addressed with sequential approaches,[37] and the overall problem is solved in a loop via a sequence of MILPs and PDOs.[38] On the other hand, in the simultaneous collocation method, all input and state variables in DAEs are discretized over finite elements and this results in a large set of sparse nonlinear equations that are tractable for nonlinear solvers.[39] By this means, MIDOs are converted to MINLPs.

In this study, we obtain reformulated MINLPs that can be highly nonconvex with the procedure depicted in Figure 2. For a nonconvex MINLP, the global optimum is usually very expensive to compute, even for medium-scale applications. Therefore, we accept good local solutions identified by local algorithms within reasonable solution times. The method of choice here is the nonlinear branch and bound (B&B) approach. In addition, we develop an initialization phase to accelerate the optimization algorithm: it first starts with obtaining the optimal solution of the recipe-based case (often MILP), then a continuous nonlinear optimization of the integrated model is carried out via fixing production sequences (binary variables) to the recipe solutions. Finally, the integrated model is solved departing from the solution generated by the previous step that is consistent with dynamic models with controls fixed.

Case studies

We test the integrated formulation in comparison with the recipe-based approach to show the benefits of incorporating dynamics into scheduling. In the recipe-based approach, batch recipes are given in advance which contain fixed values of control inputs and processing times of operating states. With these, we reduce the integrated models to their equivalent linear counterparts. The reduced models are recast and solved as MILPs (see the appendix for details) to determine the corresponding optimal schedules for the recipe-based case. Nonetheless, the integrated models maintain full degrees of freedom in control, covering detailed dynamic and quality information. In this work, we create all the models in the GAMS[40] environment and apply CPLEX[41] as the MILP solver, while SBB[42] is used to solve MINLPs using

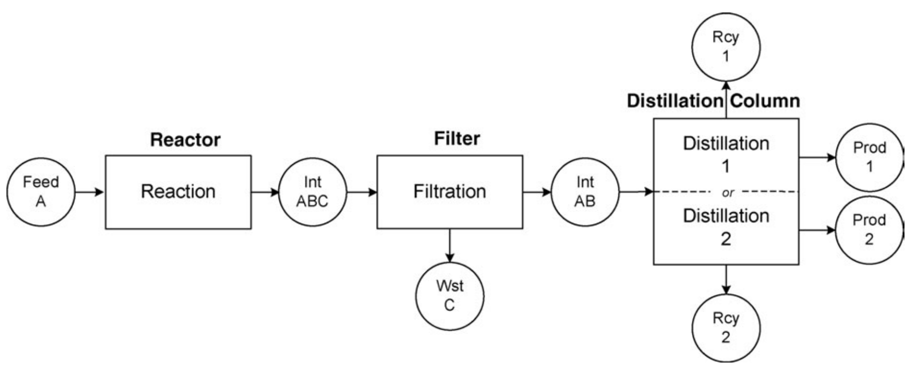
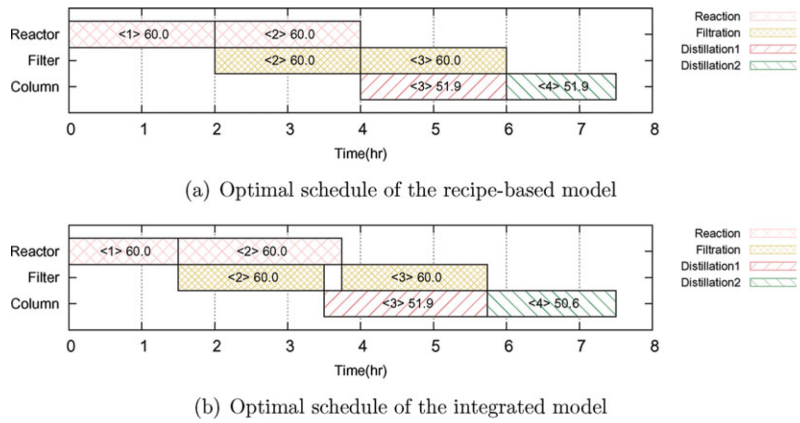


Figure 3. State equipment network for the flowshop plant.



(a) Optimal schedule of the recipe-based model

(b) Optimal schedule of the integrated model

Figure 4. Gantt charts of the flowshop plant case.

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

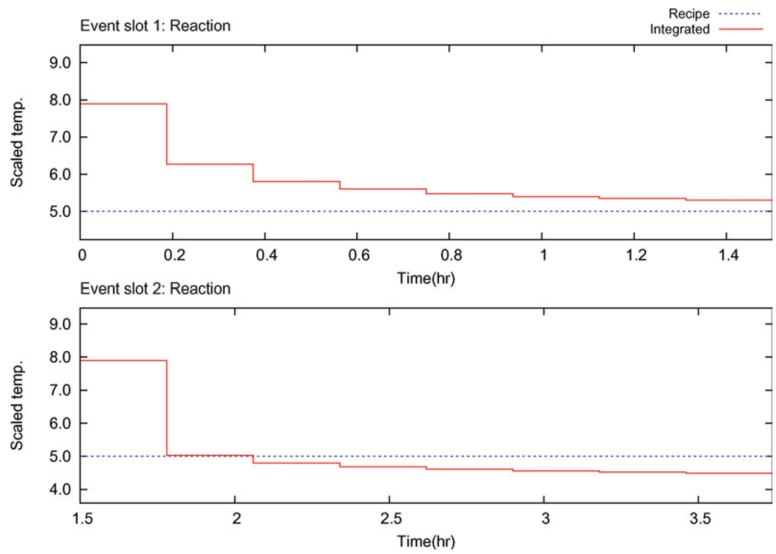


Figure 5. Optimal operating profiles for Reactor.

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

CONOPT[43] as the nonlinear subsolver. All these calculations are performed on a 64 bit Windows system with Intel Core i7 CPU 2.8 GHz and 9 GB RAM. Two case studies are considered. In the first, a simple three-stage two-product process is studied; next, we test the integrated approach on a more complex jobshop plant. These two examples are typical batch processes and the models are based on typical batch unit models such as in Bhatia and Biegler,[12] as well as other literature references. For detailed information including scheduling parameters and unit dynamic models, please refer to the appendix.

A Flowshop Plant. The flowshop plant in this case study consists of a batch reactor, a filter and a distillation column. Two species of products that differ only in purity are produced following the same three-stage procedure described by the following material and operating states

$$\text{FeedA} \xrightarrow{\text{Reaction}} \text{IntABC} \underset{\rightarrow \text{WasteC}}{\xrightarrow{\text{Filtration}}} \text{IntAB} \underset{\rightarrow \text{Recycle1}}{\xrightarrow{\text{Distillation1}}} \text{Product1}$$

$$\text{FeedA} \xrightarrow{\text{Reaction}} \text{IntABC} \underset{\rightarrow \text{WasteC}}{\xrightarrow{\text{Filtration}}} \text{IntAB} \underset{\rightarrow \text{Recycle2}}{\xrightarrow{\text{Distillation2}}} \text{Product2}$$

To represent the flow shop using the SEN framework, there need to be three units (Reactor, Filter, Distillation Column) with four operating states (Reaction, Filtration, Distillation 1, Distillation 2) and eight material states (FeedA, IntABC, IntAB, WstC, Rcy1, Rcy2, Prod1, Prod2) in the graph as shown in Figure 3. For the two distillation states, Distillation 1 lasts longer with a smaller constant reflux ratio. Thus, distillate Product 2 is purer than Product 1. As a typical reaction-separation process, reaction and distillation states are recognized as the key stages where process dynamics can expect considerable benefits, so we embed dynamic models for all these stages and operate the filter without optimization of the control.

The time horizon of interest lasts for 7.5 hours, and Product 1 and 2 are produced once for each within the time window. The number of event slots are determined by running the recipe-based MILP model, where the total number of event slots increases from a small value until no improvement on the objective can been achieved by adding more slots. In the integrated formulation, the same number of slots is used to align with the recipe-based model,

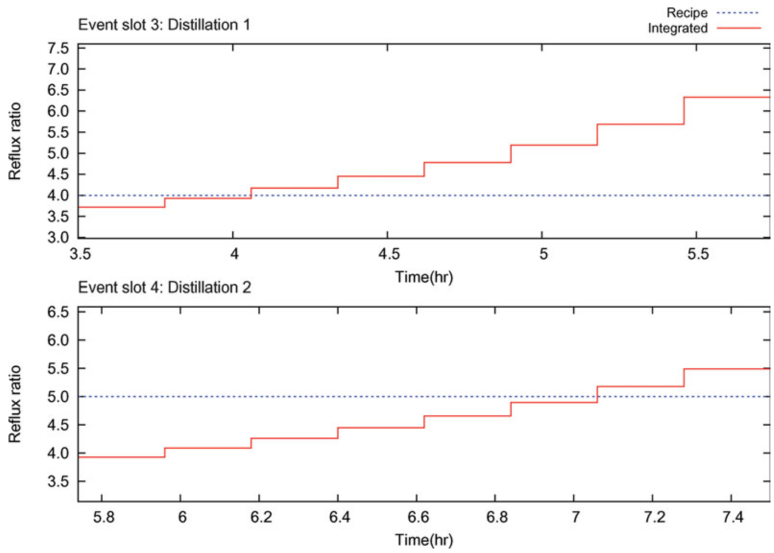


Figure 6. Optimal operating profiles for Distillation Column.

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

though a larger number may improve the optimum. In this example, both models adopt four event points to capture the sequencing behavior. Besides, the integrated model provides both discrete-time and continuous-time control strategies for operational decisions. To approximate the dynamics accurately, eight finite elements are used inside each event slot, along with a two-point Radau collocation scheme.

We solve both the integrated and the recipe-based formulations and draw the optimal schedules as Gantt charts in Figure 4 for comparison. Active operating states are depicted as patterned slots. In each slot, the number within brackets indicates the corresponding event number, and the following number is the batch size. In this example, the sequences of operations are the same for both formulations, but the timings are dissimilar. In both optimal schedules, the reaction and the distillation states are executed twice. However in the integrated case, the processing times show run-to-run variability in both dynamic operating states. In addition, the optimal control profiles of the two dynamic units are drawn in Figures 5 and 6. The dotted lines sketch the simple operating strategy specified by the recipe, while the solid lines represent the optimal dynamic curves obtained from the integrated model. It can be seen that an operating state may be executed differently in its individual appearances in the integrated model, such that the degrees of freedom in unit operations are fully explored by this method.

Some important data from the model statistics and solution are listed in Table 1. Because the same number of event slots is used, the two models have equal numbers of binary variables. However, the integrated model requires an additional group of continuous variables and constraints to account for the process dynamics that the recipe model has not taken into account. These variables and constraints consequently increase the problem size and bring in nonlinearity and nonconvexity. Since the scale of this example is rather small, both models can be solved fully such that the gap between the original problem and its relaxation is closed. In the recipe case, the optimum is even found at the presolve stage by CPLEX. On the other hand, SBB explores 88 nodes and finds the best solution at node 64 for the integrated model. In terms of the objective value, the net profit gained by the integrated approach is 36.0% higher. This improvement of profitability is very promising, though additional control devices may need to be invested to implement the dynamic control strategies. As an outcome of the integrated decision-making approach, the reactor and the column are operated cooperatively. For instance, in the manufacturing sequence of Product 2, the reactor prepares a purer intermediate material compared with that of the recipe-based model by using a longer processing time (2.24(hr) vs. 2(hr)*) and a dynamic temperature curve (Figure 5). To be more specific, after removing waste component *C* by the filter, the purity levels of intermediate product IntAB are 90.7% and 89.5% (the concentration of component *B*) for the integrated and the recipe-based approaches, respectively. As a consequence, the distillation column is able to run with a 17.3% longer processing time and a reflux profile that has a smaller average value than the recipe value (see Figure 6), but the products still meet the quality requirement on purity, which is at least 99.7% for both cases. In the integrated solution, more Product 2 can be obtained from the distillation (26.3(kg) vs. 21.3(kg)), even though the batch size of the distillation operation is slightly less than that of the recipe-based schedule (see Figure 4). To account for the manufacturing costs, the same amount of raw materials are consumed in both cases,

Table 1. Comparison of Results for the Flowshop Example

		Statistics		
Model	Type	Var.(Discrete) No.	Nonlinear Var. No.	Cons. No.
Recipe-based	MILP	153 (16)	0	499
Integrated	MINLP	1789 (16)	1240	4338

		Solution		
Model	Profit (*MU*)	CPU time (s)	Node (Best) No.	Gap (%)
Recipe-based	407	0.11	0 (0)	0.0
Integrated	554	46.92	88 (64)	0.0

*The former value comes from the integrated solution and the latter is from the recipe-based solution, and the following pairs follow the same order.

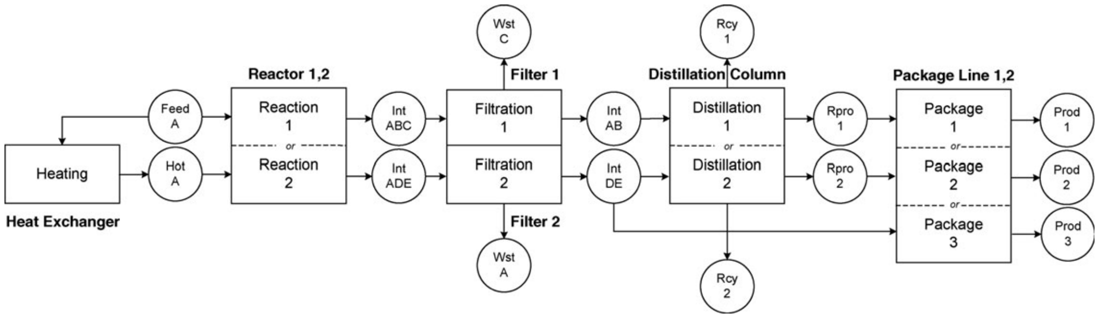
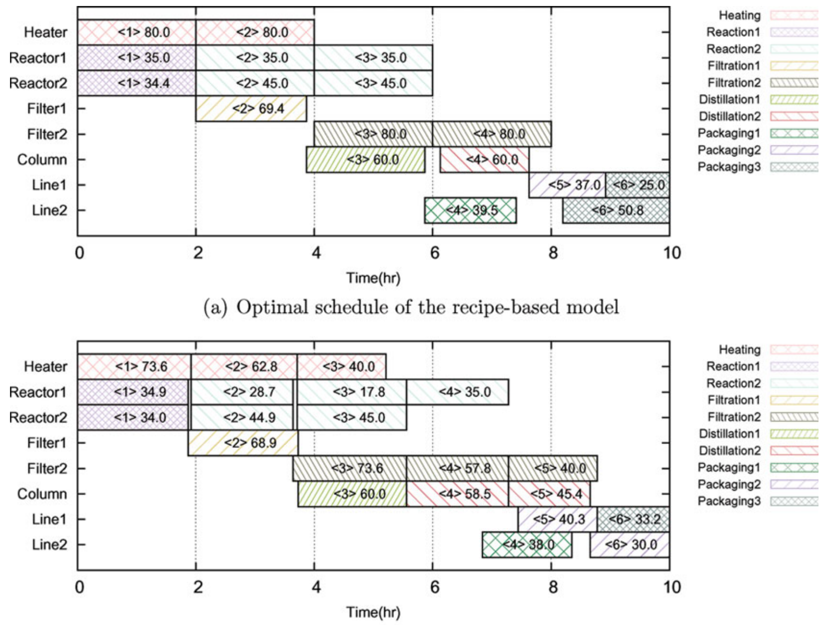


Figure 7. State equipment network for the jobshop plant.



(a) Optimal schedule of the recipe-based model

(b) Optimal schedule of the integrated model

Figure 8. Gantt charts of the jobshop plant case.

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

while the operating costs are higher in the integrated case (512(*MU*) vs. 457(*MU*)[†]). In this example however, the increment in product sales more than compensates for the increased cost. There are probably other opportunities for trade-offs between product sales and manufacturing costs by adjusting operational level and scheduling level decisions. The integrated approach is able to take advantage of these opportunities, and therefore improves the profitability of the overall process considerably.

A Jobshop Plant. In a jobshop plant, products follow dissimilar routes of processing steps such that sequencing is no longer trivial for a scheduler. In this example, the set of units in the plant contains one heat exchanger, two reactors, two filters, one distillation column, and two packaging lines. Three different varieties of products are produced via procedures described by the following material and operating states.

$$\text{FeedA} \xrightarrow{\text{Reaction1}} \text{IntABC} \xrightarrow[\to\text{WasteC}]{\text{Filtration1}} \text{IntAB} \xrightarrow[\to\text{Recycle1}]{\text{Distillation1}} \text{Rprod1} \xrightarrow{\text{Packaging1}} \text{Product1}$$

$$\text{FeedA} \xrightarrow{\text{Heating}} \text{HotA} \xrightarrow{\text{Reaction2}} \text{IntADE} \xrightarrow[\to\text{WasteA}]{\text{Filtration2}} \text{IntDE} \xrightarrow[\to\text{Recycle2}]{\text{Distillation2}} \text{Rprod2} \xrightarrow{\text{Packaging2}} \text{Product2}$$

$$\text{FeedA} \xrightarrow{\text{Heating}} \text{HotA} \xrightarrow{\text{Reaction2}} \text{IntADE} \xrightarrow[\to\text{WasteA}]{\text{Filtration2}} \text{IntDE} \xrightarrow{\text{Packaging3}} \text{Product3}$$

Both reactors operate Reaction 1 and 2, and similarly, the packaging lines and the distillation column work for multiple purposes, but the heater and the filters have dedicated functions. Again, we embed dynamic models for the reaction and distillation states. The SEN representation of this problem is shown in Figure 7. In this figure, parallel units that share the

[†]These numbers are the total operating costs for the production sequence.

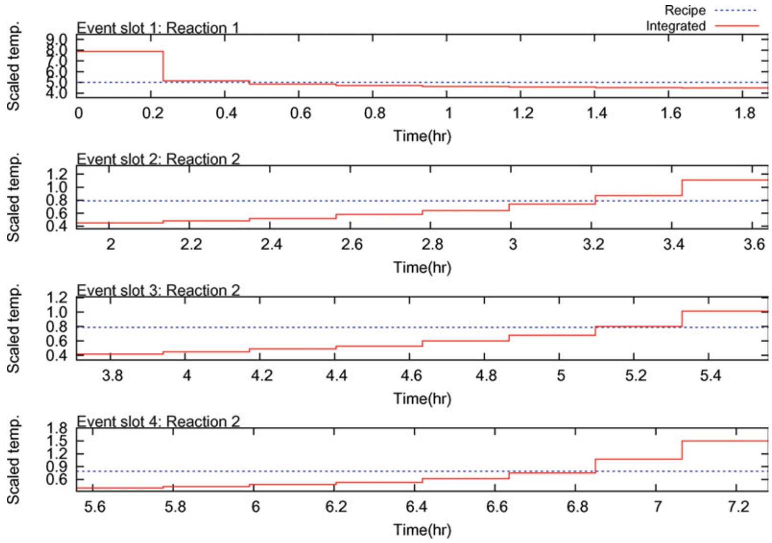


Figure 9. Optimal operating profiles for Reactor 1.

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

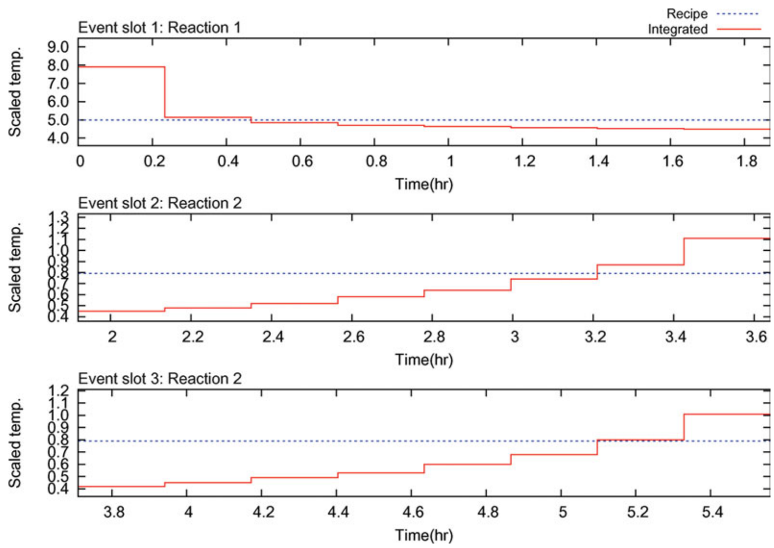


Figure 10. Optimal operating profiles for Reactor 2.

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

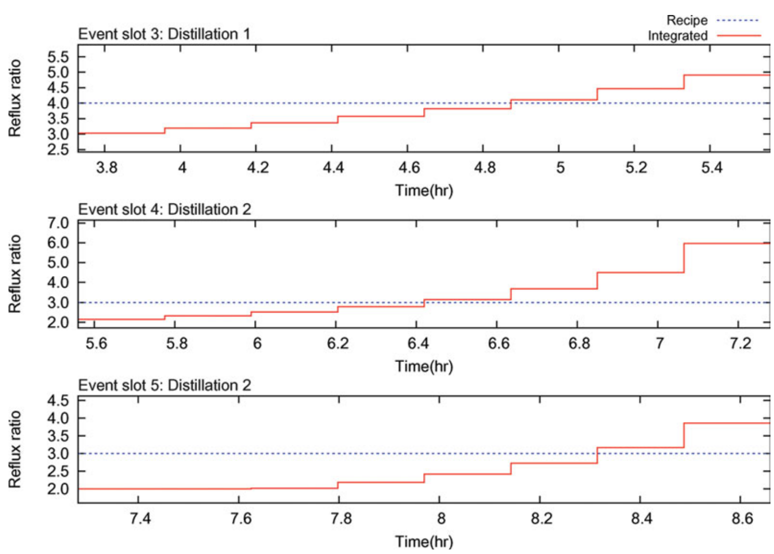


Figure 11. Optimal operating profiles for Distillation Column.

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

Table 2. Comparison of Results for the Jobshop Example

Model	Type	Statistics Var.(Discrete) No.	Nonlinear Var. No.	Cons. No.
Recipe-based	MILP	676 (90)	0	1079
Integrated	MINLP	4978 (90)	2292	12,507

Model	Profit (*MU*)	Solution CPU time (s)	Node (Best) No.	Gap (%)
Recipe-based	1374	0.366	288 (199)	0.0
Integrated	1935	9564	5000 (1602)	67.9

same set of operating states are depicted in a combined manner, since the possible connections are the same. In this example, schedules are designed to predict a time horizon of 10 hours, within which six event points are defined. The optimal schedules of the recipe-based and the integrated approaches are compared in Figure 8. The optimal operating profiles of the three dynamic units are depicted in Figures 9–11, and important data are listed in Table 2. In the Gantt charts (Figure 8), it can be seen that a larger number of operations are executed in the integrated case, e.g., one more slot for Reaction 2 of Reactor 1. In the recipe-based schedule, Packaging 2 occurs only once and Packaging 3 takes place twice; but it is the opposite in the integrated schedule so that the production of Product 2 is much higher in this case. As described earlier, the production sequence of Product 2 requires an additional distillation operation than that of Product 3. As a result, Product 2 is purer (Product 2: 99.3% vs. Product 3: 74.0%) and of a higher market price (Product 2: 50(*MU*/*kg*) vs. Product 3: 20(*MU*/kg)). The integrated schedule leads to a better production of Product 2 that consequently contributes to the increase of profit (see Table 2), and one should notice that this schedule is not optimal nor even feasible for the recipe scheme. The result here shows when controls are allowed to vary, the design of batch schedules becomes more flexible. For this reason, one may obtain a product portfolio that differs from the recipe-based method, and the portfolio can be more profitable as well.

There are three units that operate with dynamic operating states in this example: Reactor 1, Reactor 2 and Distillation Column. In the optimal integrated schedule, Reactor 1 conducts Reaction 1 once and Reaction 2 three times consecutively in its first four event slots. For these four active operating states, the temperature profiles vary with time, while staying within different feasible regions due to different restrictions of operations. Again, the integrated model is much larger in size and better in profitability. In this more complex example, we observe significant increase of computational times for the integrated model as the number of discrete variables grows. Moreover, it becomes prohibitively expensive to reduce the remaining gap to a small value. During the branch and bound procedure, integral feasible incumbents are only discovered at or just above the leaf nodes, so the improvement of the lower bound is rather slow and pruning large size subtrees is hardly observed. However, SBB is able to discover good feasible solutions in reasonable times with its default tree search algorithm. For instance, in the result of this example, it finds a good solution at node 1602 after a 3022 CPUs search, and spends the rest of the time closing the gap until the maximum node limit 5000 is exceeded. From the objective function, the net profit of the integrated approach is 40.8% higher than the recipe-based model, which is even more promising than the previous flowshop case.

Conclusions

The integrated formulation for scheduling and dynamic optimization has been developed within the state equipment network framework and serves for multiproduct and multipurpose batch plants. The basic elements of the SEN we use in this work are identified as material state, equipment and operating state. With this definition, we focus on the allocation of operating states over a scheduling horizon and process units to design batch schedules and provide optimal control strategies for the execution of active operating states. In the formulation, scheduling considerations and unit operations are tied together. Besides, a number of redundant constraints on timing and mass balance are introduced for the sake of better solution performance. With this, the problem is modeled as an MLDO program to best capture the essence and practice of process integration between the site level and the unit level.

To solve the optimization problem, we convert MLDOs to MINLPs by sequentially applying Big M reformulation of disjunctions and simultaneous collocation of DAEs. The resulting problem incurs nonconvexity from combinatorial and continuous nonlinear variables. We accept satisfactory feasible solutions from branch and bound algorithms because of the low computational load, in comparison with global MINLP algorithms. To better facilitate the tree search, a good starting point is obtained via the initialization phase, where the counterpart recipe-based problem is solved. The proposed formulation is tested on flowshop and jobshop case studies and shows advantages over the recipe-based approach, because dynamic optimization of unit operations leads to better overall performance of batch units.

Future developments of this work can be elaborated in several areas. As a two-fold problem, both more sophisticated and comprehensive scheduling rules and better dynamic models demand our attention. Furthermore, solving large-scale applications remains difficult for the integrated formulation, since the resulting nonconvex MINLPs are computationally expensive. In particular, when solved as relaxed NLPs, the problem involves a large number of redundant constraints coming from inactive operating states, and these constraints burden the NLP subsolver with nonconvexity and degenerate constraints. On the other hand, branch and bound on integer variables becomes inefficient when dealing with large systems or long horizon problems, which have a considerable number of binaries. Therefore, a decomposition method for solving the integrated problem is in the scope of our further development.

Notation

Indices

$j = j \in \mathcal{J}$ units
$r = r \in \mathcal{R}$ materials
$s = s \in \mathcal{S}$ operating states
$n = n \in \mathcal{N}$ event slots/points (staring points of event slots)
$c = c \in \mathcal{C}$ chemical components
$k = k \in \mathcal{K}$ finite elements
$q = q \in \mathcal{Q}$ collocation points

Sets

$\mathcal{J}$ = units
$\mathcal{R}$ = materials

$\mathcal{R}_{prod}$ = final products
$\mathcal{R}_{raw}$ = raw materials
$\mathcal{S}$ = operating states
$\mathcal{N}$ = event slots
$\mathcal{J}_s$ = units that have operating state s
$\mathcal{J}_r^p$ = units that produce resource r
$\mathcal{J}_r^c$ = units that consume resource r
$\mathcal{R}_s^p$ = materials produced in operating state s
$\mathcal{R}_s^c$ = materials consumed in operating state s
$\mathcal{R}_j^p$ = materials produced in unit j
$\mathcal{R}_j^c$ = materials consumed in unit j
$\mathcal{S}_j$ = operating states that appear in unit j
$\mathcal{S}_r^p$ = operating states that produce material r
$\mathcal{S}_r^c$ = operating states that consume material r
$\mathcal{C}_r$ = components of material r
$\mathcal{K}$ = finite elements
$\mathcal{Q}$ = collocation points

Parameters

N = cardinality of set $\mathcal{N}$
P_r = price of material r
H = scheduling horizon
B_j^{max},B_j^{min} = upper and lower bound of batch size for unit j
E_r^{max},E_r^{min} = upper and lower bound of excess material r
$\bar{E}_r$ = initial amount of material r
h_k = length of finite element k
Ω_q = collocation coefficients in Runge-Kutta basis
$\eta_{r,c,1}$ = initial concentration distribution of r
$\bar{\eta}_{r,c}$ = quality requirement for excess material r
$\bar{\phi}_{j,r,c}$ = quality requirement for material r produced by unit j

Variables

$w_{j,s,n}$ = binary variable indicate state of unit j within event slot n
$F_{j,n}$ = operating cost of unit j within event slot n
$E_{r,n}$ = amount of excess material at event point n
$E_{r,0}$ = amount of material r at the beginning of time horizon
$R_{j,r,n}^p$ = production of r by unit j at the end of slot n
$R_{j,r,n}^c$ = consumption of r by unit j at event point n
$b_{j,n}$ = batch size of unit j in state s within event slot n
$T_{j,n}$ = beginning time of event slot n
$Tp_{j,n}$ = processing times of unit j within event slot n
$z_{j,n}$ = differential state variables of unit j within slot n
$y_{j,n}$ = algebraic state variables of unit j within slot n
$u_{j,n}$ = control variables of unit j within slot n
τ = normalized time
$\eta_{r,c,n}$ = fraction of component c in r at event point n
$\phi_{j,r,c,n}$ = fraction of component c in r produced by j at the end of n

Superscripts

min = minimum
max = maximum
p = production
c = consumption

Literature Cited

1. Chachuat B, Singer A, Barton P. Global methods for dynamic optimization and mixed-integer dynamic optimization. *Ind Eng Chem Res*. 2006;45:8373–8392.
2. Harjunkoski I, Nyström R, Horch A. Integration of scheduling and control: theory or practice? *Comput Chem Eng*. 2009;33:1909–1918.
3. Kondili E, Pantelides C, Sargent R. A general algorithm for short-term scheduling of batch operations–I. MILP formulation. *Comput Chem Eng*. 1993;17:211–227.
4. Pantelides C. Unified frameworks for optimal process planning and scheduling. In: *Proceedings on the Second conference on Foundations of Computer Aided Operations*. Cache Publications: New York, 1994; pp. 253–274.
5. Maravelias C, Grossmann I. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Ind Eng Chem Res*. 2003;42:3056–3074.
6. Castro P, Barbosa-Povoa A, Matos H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. *Ind Eng Chem Res*. 2001;40:2059–2068.
7. Shaik M, Floudas C. Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using RTN framework. *Comput Chem Eng*. 2008;32:260–274.
8. Susarla N, Li J, Karimi I. A novel approach to scheduling multipurpose batch plants using unit-slots. *AIChE J*. 2010;56:1859–1879.
9. Schilling G, Pantelides C. Optimal periodic scheduling of multipurpose plants. *Comput Chem Eng*. 1999;23:635–655.
10. Wu D, Ierapetritou M. Cyclic short-term scheduling of multiproduct batch plants using continuous-time representation. *Comput Chem Eng*. 2004;28:2271–2286.
11. Aytug H, Lawley M, McKay K, Mohan S, Uzsoy R. Executing production schedules in the face of uncertainties: A review and some future directions. *Eur J OperRes*. 2005;161:86–110.
12. Bhatia T, Biegler L. Dynamic optimization in the design and scheduling of multiproduct batch plants. *Ind Eng Chem Res*. 1996;35:2234–2246.
13. Mishra B, Mayer E, Raisch J, Kienle A. Short-term scheduling of batch processes. A comparative study of different approaches. *Ind Eng Chem Res*. 2005;44:4022–4034.
14. Ierapetritou M, Floudas C. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Ind Eng Chem Res*. 1998;37:4341–4359.
15. Oldenburg J, Marquardt W, Heinz D, Leineweber D. Mixed-logic dynamic optimization applied to batch distillation process design. *AIChE J*. 2003;49:2900–2917.
16. Terrazas-Moreno S, Flores-Tlacuahuac A, Grossmann I. Simultaneous design, scheduling, and optimal control of a methyl-methacrylate continuous polymerization reactor. *AIChE J*. 2008;54:3160–3170.
17. Wongrat W, Younes A, Elkamel A, Douglas P, Lohi A. Control vector optimization and genetic algorithms for mixed-integer dynamic optimization in the synthesis of rice drying processes *J Franklin Inst* 2010;348:1318–1338.
18. Prata A, Oldenburg J, Kroll A, Marquardt W. Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. *Comput Cheml Eng*. 2008;32:463–476.
19. Terrazas-Moreno S, Flores-Tlacuahuac A, Grossmann I. Simultaneous cyclic scheduling and optimal control of polymerization reactors. *AIChE J*. 2007;53:2301–2315.
20. Williams H. *Model Building in Mathematical Programming*. Chichester, West Sussex, England: Wiley, 1999.
21. Biegler L. An overview of simultaneous strategies for dynamic optimization. *Chem Eng Process. Process Intens*. 2007;46:1043–1053.
22. Janak S, Lin X, Floudas C. Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: Resource constraints and mixed storage policies. *Ind Eng Chem Res*. 2004;43:2516–2533.
23. Smith EM.On the optimal design of continuous processes. Ph.D. thesis, Imperial College of Science, Technology and Medicine, London, UK. 1996.
24. Smith E, Pantelides C. Design of reaction/separation networks using detailed models. *Comput Chem Eng*. 1995;19:83–88.
25. Yeomans H, Grossmann I. A systematic modeling framework of superstructure optimization in process synthesis. *Comput Chem Eng*. 1999;23:709–731.
26. Al-Qahtani K, Elkamel A. Multisite facility network integration design and coordination: An application to the refining industry. *Comput Chem Engi*. 2008;32:2189–2202.
27. Méndez C, Cerdá J, Grossmann I, 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. Floudas C, Lin X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comput Chem Eng*. 2004;28:2109–2129.
29. Shaik M, Floudas C. Improved unit-specific event-based continuous-time model for short-term scheduling of continuous processes: rigorous treatment of storage requirements. *Ind Eng Chem Res*. 2007;46:1764–1779.
30. Biegler L, Zavala V. Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization. *Comput Chem Eng*. 2009;33:575–582.
31. Deshpande S, Patwardhan S, Methekar R, Rengaswamy R. Development of a closed form nonlinear predictive control law based on a class of Wiener models. *Ind Eng Chem Res*. 2009;49:148–165.
32. Biegler L. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010.
33. Lee S, Grossmann I. New algorithms for nonlinear generalized disjunctive programming. *Comput Chem Eng*. 2000;24:2125–2142.

34. Lee S, Grossmann I. Logic-based modeling and solution of nonlinear discrete/continuous optimization problems. *Ann Oper Res*. 2005;139:267–288.
35. Li J, Floudas C. Optimal event point determination for short-term scheduling of multipurpose batch plants via unit-specific event-based continuous-time approaches. *Ind Eng Chem Res*. 2010;49:7446–7469.
36. Hui C. Optimizing chemical processes with discontinuous function—a novel formulation. *Comput Chem Eng*. 1999;23:479–482.
37. Bryson A. *Dynamic Optimization*, Menlo Park, CA: Addison-Wesley, 1999.
38. Chatzidoukas C, Kiparissides C, Perkins J, Pistikopoulos E. Optimal grade transition campaign scheduling in a gas-phase polyolefin FBR using mixed integer dynamic optimization. *Comput Aid Chem Eng*. 2003;14:71–76.
39. Biegler L. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Comput Chem Eng*. 1984;8:243–247.
40. Brooke A, Kendrick D, Meeraus A, Raman R, Rosenthal R. *GAMS a User's Guide*. GAMS Development Corporation: Washington DC, 2006.
41. *CPLEX I. 11.0 User's Manual*. ILOG SA: Gentilly, France, 2008.
42. Bussieck M, Drud A. *SBB: a new solver for mixed integer nonlinear programming*. 2000. http://www.gams.com/presentations/or01/sbb.pdf.
43. Drud A. CONOPT–a large-scale GRG code. *INFORMS J Comput*. 1994;6:207.

Appendix A: The Flowshop Plant Example

Models of operations

Reaction. The description of reaction kinetics is based on Arrhenius equation. For simplicity, all the subscripts are neglected.

Material state transition:

$$\text{FeedA} \longrightarrow \text{IntABC}$$

Reaction formula:

$$\text{A} \xrightarrow{k_1} \text{B}$$
$$\text{B} \xrightarrow{k_2} C$$

DAE system:

$$\begin{aligned} \frac{dc_A}{dt} &= -uc_A^2 & c_A(0) &= 1 \\ \frac{dc_B}{dt} &= uc_A^2 - \beta u^\alpha c_B & c_B(0) &= 0 \\ \frac{dc_C}{dt} &= \beta u^\alpha c_B & c_c(0) &= 0 \end{aligned} \tag{A1}$$

where scaled temperature u is defined to substitute rate constants k_1, and k_2, and the unit used here is m^3/(kg hr). While there is no energy balance equation, the differences in activation energy can largely affect the trends of the operating temperature as we can observe from the case study results. The costs of reactions come from the usage of heating utility of unit price p. The cost function of reactors can be described as Eq. A2.

$$F = pb \int_0^{Tp} u(t)dt \tag{A2}$$

Important parameters are tabulated as follows:

Reaction	p (MU/kg)	α	β	$T_p^{\min}$	$T_p^{\max}$ (hr)	T_p^{recipe}	$u^{\min}$	$u^{\max}$ m^3/(kg·hr)	u^{recipe}
	0.3	2	3.875×10^{-3}	1.5	3	2	1.8	7.9	5

Distillation. Material state transitions:

$$\begin{aligned} \text{Dis1}: &\quad \text{IntAB} \longrightarrow \text{RawProd1} + \text{Recycle1} \\ \text{Dis2}: &\quad \text{IntAB} \longrightarrow \text{RawProd2} + \text{Recycle2} \end{aligned}$$

The dynamic models of the distillation states are based on a simplified tray-type binary distillation column model (four trays in our study), which has been set up by making following assumptions:

- Binary distillation
- Negligible tray holdup
- Tray equilibrium with constant relative volatility α
- Equimolar overflow (constant vapor rate V)

Differential equations:

$$\begin{aligned} \dot{S} &= L - V & S(0) &= b \\ \dot{x_b} &= \frac{V(x_b - x_d)}{(R+1)S} & x_b(0) &= \eta \end{aligned} \tag{A3}$$

Constant vapor flow:

$$V = kb \tag{A4}$$

Composition balance for each tray:

$$\begin{aligned} &Lx_i + Vy_i = Lx_{i+1} + Vy_{i-1} \qquad \forall i = 2 \ldots NT-1 \\ &Lx_N + Vy_N = Lx_d + Vy_{N-1} \\ &Lx_1 + Vy_1 = Lx_2 + Vy_b \end{aligned} \tag{A5}$$

Phase equilibrium:

$$\begin{aligned} y_i &= \frac{\alpha x_i}{1 + (\alpha - 1)x_i} \qquad \forall i = 1 \ldots NT \\ y_b &= \frac{\alpha x_b}{1 + (\alpha - 1)x_b} \end{aligned} \tag{A6}$$

At the top of the column:

$$\begin{aligned} R &= \frac{L}{D} \\ L + D &= V \end{aligned} \tag{A7}$$

Average purity:

$$\bar{x_d} \int_0^t D dt = S_0 x_0 - S(t) x_b(t) \tag{A8}$$

Requirements for manufacturing:

$$\begin{aligned} R^{\min} \le R &\le R^{\max} \\ \bar{x_d} &\ge \bar{x_d}^{\min} \end{aligned} \tag{A9}$$

The cost function is defined as below in Eq. A10.

$$F = pVT_p \tag{A10}$$

According to this model, x_b is always decreasing with time. Hence, to maintain a high product purity, the reflux ratio profiles are generally increasing with time as we can see in the case study results. The differences between the two operating state models lie in some process indices listed in the following table.

Distillation	p $MU/(\mathrm{m}^3\cdot\mathrm{hr})$	α	k m^3/kg	$T_p^{\min}$	$T_p^{\max}$ (hr)	T_p^{recipe}	$R^{\min}$	$R^{\max}$	R^{recipe}
1	1.5	2.46	1.646	1.5	3	2	2	7	4
2	1.5	2.46	1.646	1.125	2.5	1.5	2	7	5

Filtration. The goal of a filtration operation is to completely remove waste components from the resultants of its antecedent reaction operation. We assume the filter is operated on the basis of established recipes, where processing times are approximated as linear functions of batch sizes. Besides, productions and consumptions are algebraic functions of the amount and the composition of feed materials. If the preceding reaction is operated identically, then the feed composition can be treated as a constant. Otherwise, it remains a variable and the filtration model is nondynamic but nonlinear.

Material state transitions:

$$\text{IntABC} \longrightarrow \text{IntAB} + \text{WasteC}$$

Production and consumption:

$$\begin{aligned} R^{\mathrm{p}}_{\mathrm{IntAB}} &= (1-\eta_{\mathrm{IntABC},C})b \\ R^{\mathrm{p}}_{\mathrm{WstC}} &= \eta_{\mathrm{IntABC},C}b \\ R^{\mathrm{c}}_{\mathrm{IntABC}} &= b \end{aligned} \tag{A11}$$

Operating cost:

$$F = R^{\mathrm{p}}_{\mathrm{IntAB}} + 4R^{\mathrm{p}}_{\mathrm{WstC}} \tag{A12}$$

Reduced MILP Formulation. The recipe-based models we use to compare with the integrated formulation are linear without dynamic variables, quality variables or corresponding constraints. The reduced problem can be reformulated as an MILP and solved for the same objective function as the integrated formulation. Here, instead of the original disjunctive DAEs, unit operations are described through Eq. A13.

$$\cdots \begin{bmatrix} w_{j,s,n} = 1 \\ R^{\mathrm{p}}_{j,r,n} = \mu^{\mathrm{p}}_{j,s} b_{j,n} \quad (\forall r \in \mathcal{R}^{\mathrm{p}}_s) \\ R^{\mathrm{c}}_{j,r,n} = \mu^{\mathrm{c}}_{j,s} b_{j,n} \quad (\forall r \in \mathcal{R}^{\mathrm{c}}_s) \\ R^{\mathrm{p}}_{j,r,n} = 0 \quad (\forall r \notin \mathcal{R}^{\mathrm{p}}_s) \\ R^{\mathrm{c}}_{j,r,n} = 0 \quad (\forall r \notin \mathcal{R}^{\mathrm{c}}_s) \\ Tp_{j,n} = \alpha_{j,s} + \beta_{j,s} b_{j,n} \\ F_{j,n} = \gamma_{j,s} b_{j,n} \end{bmatrix}_{s \in \mathcal{S}_j} \cdots \bigvee \begin{bmatrix} \sum_{s \in \mathcal{S}_j} w_{j,s,n} = 0 \\ R^{\mathrm{p}}_{j,r,n} = 0 \\ R^{\mathrm{c}}_{j,r,n} = 0 \\ Tp_{j,n} = 0 \\ F_{j,n} = 0 \end{bmatrix} \quad \forall j \in \mathcal{J}, n \in \mathcal{N} \tag{A13}$$

The parameters ($\alpha,\beta,\gamma,\mu^{\mathrm{p}},\mu^{\mathrm{c}}$) are obtained via simulation based on predefined recipes which in this study we determine manually. For each sequence of operations to make a product, we start with simulating the first stage with known controls, processing times and initial conditions according to the recipe of the operation. Then the procedure is repeated for the subsequent stages, using the information from the recipe and the calculated precedent stage. Parameters for this example are listed as follows:

	Rct	Fil	Dis1	Dis2
α(hr)				
Reactor	2			
Filter		0.8		
Column			2	1.5
β(hr/kg)				
Reactor	0			
Filter		0.02		
Column			0	0
γ(MU/kg)				
Reactor	3			
Filter		1.405		
Column			4.938	3.704
$\mu^{\mathrm{p/c}}(+/-)$				
FeedA	−1			
IntABC	1	−1		
IntAB		0.865	−1	−1
WstC		0.135		
Rcy1			0.342	
Rcy2				0.589
Prod1			0.658	
Prod2				0.411

Scheduling parameters. Material information:

Material	Component A	B	C	$\bar{\eta}$	Parameter P (MU/kg)	E_0 (kg)	$E^{\max}$ (kg)	$E^{\min}$ (kg)
FeedA	✓				5	var	400	0
IntABC	✓	✓	✓		0	0	400	0
IntAB	✓	✓			0	0	400	0
WstC			✓		0	0	400	0
Prod1	✓	✓		$B \geq 99.5\%$	30	0	400	0
Prod2	✓	✓		$B \geq 99.7\%$	45	0	400	0
Rcy1	✓	✓			0	0	400	0
Rcy2	✓	✓			0	0	400	0

Equipment information:

Unit	Operating States	$B^{\max}$ (kg)	$B^{\min}$ (kg)
Reactor	Rct	60	30
Filter	Fil	60	30
Column	Dis1,Dis2	60	30

Appendix B: The Jobshop Plant Example

Models of operations

Reaction. In this jobshop example, Reaction 1 is the same as the reaction operation in the flowshop case, and dynamic model for Reaction 2 can be written out similarly, wherein the unit of rate constants is per hour.

Material state transition:

$$\text{HotA} \longrightarrow \text{IntADE}$$

Reaction formula:

$$\mathrm{A} \xrightarrow{k_3} \mathrm{D}$$

$$\mathrm{A} \xrightarrow{k_4} \mathrm{E}$$

DAE system:

$$\begin{aligned} \frac{dc_A}{dt} &= -(u+\beta u^{\alpha})c_A & c_A(0) &= 1 \\ \frac{dc_D}{dt} &= uc_A & c_D(0) &= 0 \\ \frac{dc_E}{dt} &= \beta u^{\alpha} c_A & c_E(0) &= 0 \end{aligned} \tag{B1}$$

Important process data:

Reaction 2	p (MU/kg)	α	β	$T_p^{\min}$	$T_p^{\max}$ (hr)	T_p^{recipe}	$u^{\min}$	$u^{\max}$ (/hr)	u^{recipe}
	3	2	0.5	1.5	3	2	0.3	1.5	0.7

Distillation. The dynamic models are in the same form as the flowshop case with different constituents of the binary mixtures, and parameters for Distillation 1 and 2 are listed as follows:

Distillation	p $MU/(\text{m}^3\cdot\text{hr})$	α	k (m^3/kg)	$T_p^{\min}$	$T_p^{\max}$ (hr)	T_p^{recipe}	$R^{\min}$	$R^{\max}$	R^{recipe}
1	1.5	2.46	1.646	1.5	3	2	2	7	4
2	1.5	3.20	1.646	1.125	2.5	1.5	2	7	3

Filtration. Filtration 1 is the same as the flowshop case. For Filtration 2:

Material state transition:

$$\text{IntADE} \longrightarrow \text{IntDE} + \text{WasteA}$$

Production and consumption:

$$\begin{aligned} R^{\text{p}}_{\text{IntDE}} &= (1-\eta_{\text{IntADE},A})b \\ R^{\text{p}}_{\text{WstA}} &= \eta_{\text{IntADE},A}b \\ R^{\text{c}}_{\text{IntADE}} &= b; \end{aligned} \tag{B2}$$

Operating cost:

$$F = R^{\text{p}}_{\text{IntDE}} + 4R^{\text{p}}_{\text{WstA}} \tag{B3}$$

Heating and Packaging. These operations are described via linear equations based on mass balance. Information on determining processing times and operating costs can be found in the reduced MILP formulation, discussed in the following section.

Reduced MILP formulation

Unit operations are described through Eq. A13, where the parameters are listed as follows:

	Hting	Rct1	Rct2	Fil1	Fil2	Dis1	Dis2	Pck1	Pck2	Pck3
α(hr)										
Heater	1									
Reactor1		2	2							
Reactor2		2	2							
Filter1				1						
Filter2					1					
Column						2	1.5			
Line1								0.667	0.667	0.667
Line2								0.667	0.667	0.667
β(hr/kg)										
Heater	0.0125									
Reactor1		0	0							
Reactor2		0	0							
Filter1				0.0125						
Filter2					0.0125					
Column						0	0			
Line1								0.0167	0.0167	0.0167
Line2								0.0223	0.0223	0.0223
γ(MU/kg)										
Heater	1									
Reactor1		3	4.2							
Reactor2		3	4.2							
Filter1				1.405						
Filter2					1.453					
Column						4.938	3.704			
Line1								1	1	1
Line2								0.9	0.9	0.9
$\mu^{\text{p/c}}$(+/−)										
FeedA	−1	−1								
HotA	1		−1							
IntABC		1		−1						
IntADE			1		−1					
IntAB				0.865		−1				
IntDE					0.849		−1			−1
WstC				0.135						
WstA					0.151					
RProd1						0.658		−1		
RProd2							0.617		−1	
Rcy1						0.342				
Rcy2							0.383			
Prod1								1		
Prod2									1	
Prod3										1

Scheduling parameters

Material information:

	Component						Parameter			
Material	A	B	C	D	E	$\bar{\eta}$	P (MU/kg)	E_0(kg)	E^{max}(kg)	E^{min}(kg)
FeedA	✓						5	*var*	400	0
HotA	✓						0	0	400	0
IntABC	✓	✓	✓				0	0	400	0
IntAB	✓	✓					0	0	400	0
IntADE	✓			✓	✓		0	0	400	0
IntDE				✓	✓		0	0	400	0
WstA	✓						0	0	400	0
WstC			✓				0	0	400	0
Rprod1	✓	✓					0	0	400	0
Rcy1	✓	✓					0	0	400	0
Rprod2				✓	✓		0	0	400	0
Rcy2				✓	✓		0	0	400	0
Prod1	✓	✓				$B \geq 99.5\%$	30	0	400	0
Prod2				✓	✓	$D \geq 99.3\%$	50	0	400	0
Prod3				✓	✓	$D \geq 74.0\%$	20	0	400	0

Equipment information:

Unit	Operating states	B^{max}(kg)	B^{min}(kg)
Heater	Heating	80	40
Reactor1	Rct1, Rct2	35	17.5
Reactor2	Rct1, Rct2	45	22.5
Filter1	Fil1	80	40
Filter2	Fil2	80	40
Column	Dis1, Dis2	60	30
Line1	Pck1, Pck2, Pck3	50	25
Line2	Pck1, Pck2, Pck3	60	30

Manuscript received Oct. 4, 2011 and revision received Dec. 26, 2011.