Integration of scheduling and control with online closed-loop implementation: Fast computational strategy and large-scale global optimization algorithm

Yunfei Chu, Fengqi You*

Department of Chemical & Biological Engineering

Northwestern University

Evanston, IL 60208, USA

Abstract

In this paper, we propose a novel integration method to solve the scheduling problem and the control problem simultaneously. The integrated problem is formulated as a mixed-integer dynamic optimization (MIDO) problem which contains discrete variables in the scheduling problem and constraints of differential equations from the control problem. Because online implementation is crucial to deal with uncertainties and disturbances in operation and control of the production system, we develop a fast computational strategy to solve the integration problem efficiently and allow its online applications. In the proposed integration framework, we first generate a set of controller candidates offline for each possible transition, and then reformulate the integration problem as a simultaneous scheduling and controller selection problem. This is a mixed-integer nonlinear fractional programming problem with a non-convex nonlinear objective function and linear constraints. To solve the resulting large-scale problem within sufficiently short computational time for online implementation, we propose a global optimization method based on the model properties and the Dinkelbach's algorithm. The advantage of the proposed method is demonstrated through four case studies on an MMA polymer manufacturing process. The results show that the proposed integration framework achieves a lower cost rate by than the conventional sequential method, because the proposed framework provides a better tradeoff between the conflicting factors in scheduling and control problems. Compared with the simultaneous approach based on the full discretization and reformulation of the MIDO problem, the proposed integration framework is computationally much more efficient, especially for large-

* Corresponding author. Tel: +1 847 467 2943, fax: +1 847 491 3728; Email: you@northwestern.edu

scale cases. The proposed method addresses the challenges in the online implementation of the integrated scheduling and control decisions by globally optimizing the integrated problem in an efficient way. The results also show that the online solution is crucial to deal with the various uncertainties and disturbances in the production system.

Keywords

Cyclic scheduling, PI control, Global optimization, Decomposition algorithm, MMA polymerization process

1. Introduction

Integration of scheduling and control achieves a better overall performance than the conventional sequential approaches because a good tradeoff can be made between the two problems (Grossmann, 2005; Harjunkoski et al., 2009; Munoz et al., 2011). The integrated problem is a mixed integer dynamic optimization (MIDO) problem (Allgor & Barton, 1999; Bansal et al., 2003), which might be computationally expensive. As a consequence, most integration methods can only be applied offline. However, online implementation is critical to deal with various uncertainties and disturbances in the production system. Yet the online solution is rather more challenging because there are stringent requirements on the computational efficiency. It is the goal of the paper to propose a novel framework for integrating scheduling and control along with a fast computational strategy capable for online applications.

Different from most existing integration methods where the trajectory of the control variable (the process input) is determined directly in the integrated problem, a parameterized controller is considered in the proposed method. The controller designs, rather than the values of the control variables, are determined simultaneously with the scheduling decisions. The direct calculation of the control variable will incur the risk of open loop if the control variable cannot be updated immediately by solving the integrated problem in the sampling period. In contrast, the proposed integration structure is more robust since there is always a closed-loop controller working in real time even if the controller parameters are not updated by the integrated problem. Due to the existence of the real-time controller, the integrated problem does not need to be solved in real time, but can be solved in a relatively large time scale, e.g. once per production period. The most popular parameterized controller in the process industry is the PI controller (Astrom et al., 1998; Skogestad, 2003) and it is adopted in this work.

Efficient solution to the formulated MIDO problem is the key to the online implementation of the integrated method. Therefore, we propose an efficient computational strategy, which significantly reduces the model complexity by decomposing the dynamic optimization for the controller design from the scheduling model. For each potential transition, the range of the transition time is discretized by a number of sampling points and a controller is designed to minimize the transition cost at each transition time point. A set of controller candidates is then calculated and stored. Based on the controller candidates generated offline, the integrated MIDO problem is approximated by a simultaneous scheduling and controller selection problem, which is a mixed-integer nonlinear fractional programming problem. The objective function of this problem is the ratio of a convex quadratic function to a linear function and all constraints are linear, because the equations related to the dynamic system are all removed after the decomposition. The derived problem has a much smaller size than the original integrated problem and it can be globally optimized using a fast computational method based on the Dinkelbach's algorithm.

The novelties in the proposed integration framework are summarized as

- A novel MIDO framework for scheduling and control which calculates the controller parameters in the integrated problem rather than the direct value of the control variable.
- A fast computational strategy based on the decomposition of control problems from the integrated problem where all dynamic optimization problems are solved offline.
- An efficient global optimization algorithm based on the model properties and Dinkelbach's algorithm.

The advantage of the proposed method is demonstrated through four case studies. The proposed method based on the Dinkelbach's algorithm is compared with the solvers of BARON (Tawarmalani & Sahinidis, 2004), SBB (Bussieck & Drud, 2001), and DICOPT (Kocis & Grossmann, 1989). The results show that the proposed method globally optimizes the simultaneous scheduling and controller selection problem with the shortest computational time.

Comparisons with the traditional sequential method show that the proposed integration method can reduce the overall production cost rate by 19.7% because it provides a better balance between the inventory cost rate and the transition cost rate. Comparing with another integration method which solves the MIDO using the simultaneous approach, the proposed method is computationally much more efficient, especially for cases with a large number of products. The proposed method addresses the challenges in the online implementation of the integrated scheduling and control problems. The results also show that the online solution allows the production system to quickly respond to various uncertainties and disturbances.

The paper is organized as follows. The literature review of the existing integration method is presented in Section 1.1. Section 2 provides the formulation of the integrated problem of scheduling and control. The detailed solution strategy is presented in Section 3. Case studies of a polymer manufacturing processes are presented to demonstrate the proposed integration framework in Section 4. The conclusion is given in Section 5.

1.1 Literature review of existing integration methods

Popular methods for the offline MIDO problem can be cast into two categories: the simultaneous approach and the decomposition approach. The simultaneous approach fully discretizes the differential equations in the control problem (Biegler, 2007; Cuthrell & Biegler, 1987). In this way, the MIDO is reformulated into a mixed integer nonlinear programing (MINLP) problem. There are a number of general-purpose MINLP algorithms, including the outer approximation method (Duran & Grossmann, 1986), the branch and bound algorithm (Quesada & Grossmann, 1992), the cutting plane method (Westerlund et al., 1998), as well as deterministic global optimization algorithms, such as the branch-and-reduce method (Tawarmalani & Sahinidis, 2005) and the GloMIQO algorithm (Misener & Floudas, 2012). The simultaneous approach has been applied to scheduling problems of continuous processes (Flores-Tlacuahuac & Grossmann, 2006, 2011; Mitra et al., 2010; Terrazas-Moreno et al., 2008) and batch processes (Mishra et al., 2005; Nie et al., 2012). The simultaneous approach is straightforward. However, the integrated problem includes multiple dynamic systems (one in each transition period), so the reformulated MINLP contains a great number of nonlinear equations. The complexity in the reformulated problem incurs difficulties in solution feasibility and computational efficiency (Capon-Garcia et al., 2011; Harjunkoski et al., 2009).

An alternative approach to the integrated MIDO problem is the decomposition method (Allgor & Barton, 1999). The MIDO problem is first decomposed into a master problem and a primal problem, and then solved by iterations between the two problems. In the primal problem the dynamic optimization problem is solved with sequence variables fixed in the master problem to provide an upper bound of the optimal solution. The master problem, derived from the relaxation of the scheduling problem, is solved with the updated dynamic trajectories to provide a lower bound of the optimal solution. The iterations stop when the gap between the upper bound and the lower bound is under the given tolerance level. The advantage of the decomposition approach is that the dynamic optimization problem is separated from the scheduling problem. Besides the full discretization method (Biegler, 2007; Cuthrell & Biegler, 1987), the dynamic optimization problem can be solved by alternative algorithms, including the dynamic programming (Luus, 1990), the tracking of necessary conditions of optimality (Srinivasan et al., 2003), and the control vector parameterization (Vassiliadis et al., 1994). The decomposition approach has been applied to the polymer production processes (Nystrom et al., 2005; Nystrom et al., 2006; Prata et al., 2008). However, the lower bound provided by the master problem is often approximated, because the relaxation in the master problem is not applied to the model of the control problems and the convergence of the iterations might not be guaranteed.

Though important, online implementation of the integrated problem is addressed by only a few recent literature while most integration methods still focus on the optimal solution to open-loop control systems (Flores-Tlacuahuac & Grossmann, 2006; Nystrom et al., 2005; Nystrom et al., 2006; Terrazas-Moreno et al., 2008). Zhuge and Ierapetritou recently proposed an online method (Zhuge & Ierapetritou, 2012) that follows the idea of model predictive control and solves the integrated problem whenever a significant disturbance appears in the process. It applies the simultaneous approach and solves the resulting large-scale MINLP problem online. However, the computational demand of solving the complex MINLP in a short period, i.e. the sampling period in the control loop, can be a challenging task. Besides the integration by solving an MIDO problem, investigation of scheduling problems from the perspective of control theory also attracts increasing attentions recently (Rawlings et al., 2012; Subramanian et al., 2012).

2. Formulation of the integrated scheduling and control problem for cyclic production in CSTR

The formulation of the integrated problem of scheduling and control is presented in this section. The scheduling problem and the control problem are presented respectively in Section 2.1 and Section 2.2, and then the integrated problem is provided in Section 2.3. A widely studied production model is the cyclic manufacture of multiple products in CSTRs (Levner et al., 2010; Pinto & Grossmann, 1994; Sahinidis & Grossmann, 1991). An important instance is the grade transition in polymerization production (Embirucu et al., 1996; McAuley & Macgregor, 1992). Therefore, the focus of this paper is placed on the type of problems.

An illustration of the production process is exhibited in Fig. 1. A number of products, A, B, C, ..., are manufactured cyclically in a CSTR. The CSTR can manufacture at most one product at a time. Therefore, the production cycle is partitioned into a number of time slots and only one product is assigned to one slot (Mendez et al., 2006). Each slot is composed of two periods: the transition period and the production period. The products are manufactured only in the production periods when the CSTR operates in the steady state. The transition period represents the changeover between different products, in which the output of the CSTR varies with time.

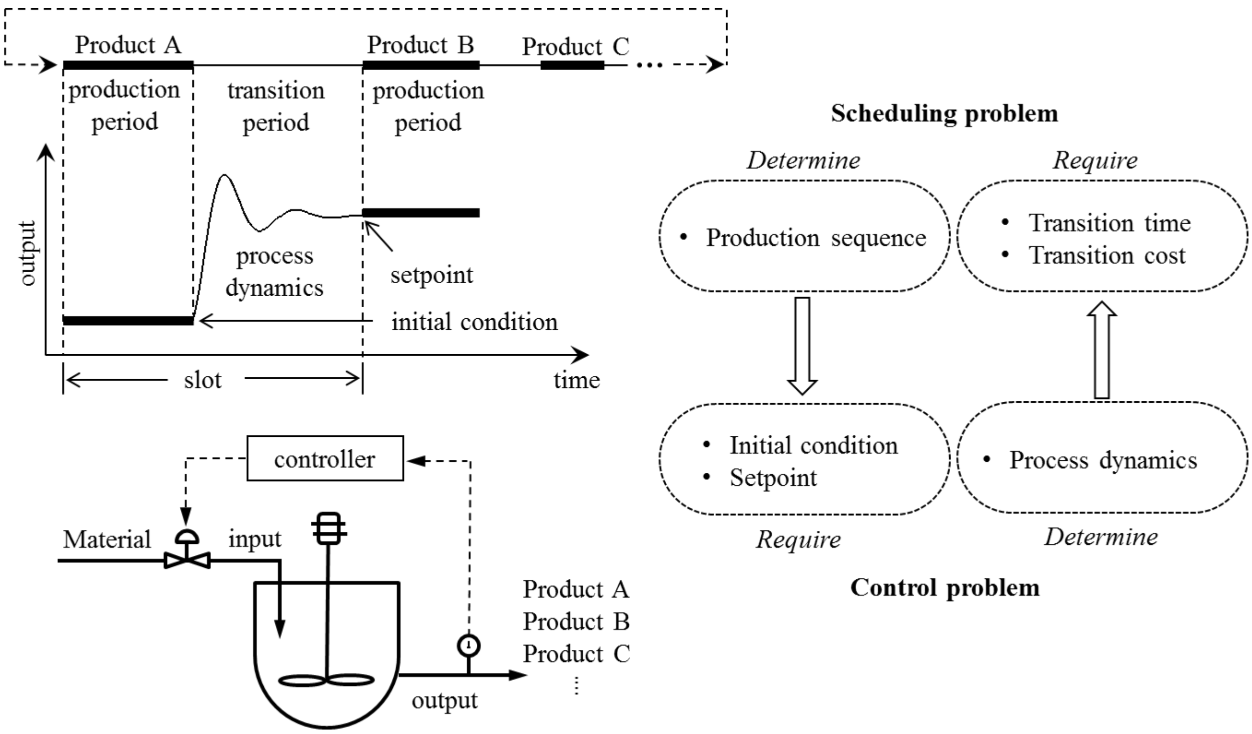


Fig 1. Cyclic production of multiple products in a CSTR.

The scheduling problem aims to minimize the production cost by determining the production sequence. The transition is controlled by a feedback controller which manipulates the process input so that the output tracks the new setpoint. It is the task of the control problem to determine the process dynamics. Obviously, the scheduling problem is coupled with the control problem. On one hand, design of the controller requires information of the initial condition and the setpoint, which is obtained from the production sequence determined by the scheduling problem. On the other hand, the scheduling decisions need the information of the transition time and the transition cost, which is obtained from the process dynamics determined by the control system. It is the coupling that motivates the development of an integration method for solving both problems simultaneously.

2.1 Cyclic scheduling problem

This subsection presents the detailed formulation of the cyclic scheduling problem, which is derived from the general case (Pinto & Grossmann, 1994). Though the proposed framework can be easily extended to the scheduling problem of parallel production lines in multiple CSTRs, the model of a single CSTR is considered in this work, because the focus is placed on the integration of scheduling and control problems rather than the solution of each individual problem.

The objective function is the total cost rate φ as

$$\min\ \varphi = \varphi_1 + \varphi_2 \tag{1}$$

where φ_1 is the inventory cost rate and φ_2 is the material cost rate. The expression of the inventory cost rate is

$$\varphi_1 = \frac{1}{t_c}\sum_{i=1}^{N}\frac{1}{2}C_i^s\left(G_i t_c - w_i\right)\Theta_i\ , \tag{2}$$

where t_c is the cycle time, C_i^s is the inventory price, G_i is the production rate of product *i*, w_i is the production amount, and Θ_i is the production time. The material cost rate is given by

$$\varphi_2 = \frac{1}{t_c}\int_0^{t_c} C^r u\left(t\right)dt\ , \tag{3}$$

where C^r is the material price, and $u(t)$ is the material flow rate. The integral calculates the accumulative consumption of the material in a cycle.

The scheduling problem includes both discrete and continuous variables. The 0-1 decision variables ξ_{ik} are introduced to denote if product *i* is assigned to slot *k*. The assignment constraints are

$$\sum_{k=1}^{N}\xi_{ik} = 1,\ \forall i\ , \tag{4}$$

$$\sum_{i=1}^{N}\xi_{ik} = 1,\ \forall k\ . \tag{5}$$

The number of time slots is assumed to be equal to the number of products, denoted by *N*. The constraint (4) indicates that each product can be manufactured only once within a production cycle while the constraint (5) implies that only one product is manufactured in each slot. It should be noted that manufacturing a product only once per cycle may not the optimal scheme. The optimal scheme should also determine how many production periods a product has in a cycle (Voudouris & Grossmann, 1993). However, one production period of a product is often assumed to simplify the solution method (Pinto & Grossmann, 1994; Sahinidis & Grossmann, 1991).

The continuous decision variables in the scheduling problem are introduced to denote the production amount of a product, the duration of each period, the production cost, and the transition cost. The amount of product *i* manufactured in a cycle is represented by

$$w_i = G_i \Theta_i, \ \forall i . \tag{6}$$

The production amount of a product needs to meet the demand from the order

$$w_i / t_c = D_i, \ \forall i , \tag{7}$$

where D_i denotes the demand rate. In the original model (Pinto & Grossmann, 1994), the demand constraint is expressed as an inequality where the production amount is required to be no less than the order demand. However, without an upper bound constraint, the production amount can be a multiple of the order demand and it is unrealistic to assume that the overproduced products can be all sold at the same price as those demanded by the order. The order driven production becomes more important to the integrated problem because a goal for the integration is to allow the production quickly track the changing environment like the rush order. Due to these reasons, the inequality constraint is modified into the equality constraint (7) where the production amount is exactly equal to the order demand. The equality constraint also helps to simply the scheduling problem in the following derivations. It should be clarified that the feasibility of the problem can always be guaranteed by tightening the inequality constraint to be an equation. Actually, the cost is minimized in the paper and the demand constraint will be automatically tightened.

The total cycle time t_c is the sum of the production time, θ_k^p, and the transition time, θ_k^t, over all time slots. It is expressed as

$$t_c = \sum_{k=1}^{N} \theta_k^p + \sum_{k=1}^{N} \theta_k^t . \tag{8}$$

According to the decomposition of the cycle time (8), the material cost in the objective function (3) is expressed as the sum of the production cost and the transition cost in each slot

$$\varphi_2 = \frac{1}{t_c} \left(\sum_{k=1}^{N} \delta_k^p + \sum_{k=1}^{N} \delta_k^t \right). \tag{9}$$

The production cost δ_k^p is the material cost integrated over the production period θ_k^p

$$\delta_k^p = \int_{\theta_k^p} C^r u(t) dt, \ \forall k . \tag{10}$$

In the production period, the CSTR operates in the steady state and the material flow rate $u(t)$ is constant

$$u(t) = \sum_{i=1}^{N} \xi_{ik} F_i, \ t \in \theta_k^p, \tag{11}$$

where F_i is the steady state value of the material flow rate for manufacturing product *i* and it is available from the production recipe. Thus, the production cost is expressed as

$$\delta_k^p = C^r \theta_k^p \sum_{i=1}^{N} \xi_{ik} F_i, \ \forall k. \tag{12}$$

The sum of the production costs over all time slots is calculated as

$$\begin{aligned} \sum_{k=1}^{N} \delta_k^p &= \sum_{k=1}^{N} C^r \theta_k^p \sum_{i=1}^{N} \xi_{ik} F_i \\ &= C^r \sum_{i=1}^{N} F_i \sum_{k=1}^{N} \theta_k^p \xi_{ik} \\ &= C^r \sum_{i=1}^{N} \Theta_i F_i \end{aligned} \tag{13}$$

where the sum of $\theta_k^p \xi_{ik}$ over all slots is equal to the production time Θ_i, i.e.

$$\Theta_i = \sum_{k=1}^{N} \theta_k^p \xi_{ik}, \tag{14}$$

because constraint (4) imposes that one and only one ξ_{ik} equals to 1 for a given product *i*.

Similar to the production cost, the transition cost δ_k^t is the material cost integrated over the transition period

$$\delta_k^t = C^r \int_{\theta_k^t} u(t)\, dt, \ \forall k. \tag{15}$$

Since the material flow rate varies in the transition period, the transition cost depends on the trajectory of the material flow rate. The trajectory is a dynamic behavior of the process and it is determined by the control system. Also, the transition time θ_k^t is determined by the control system.

If changeover products are valuable, they can be accounted for by extending the model. Besides the transition cost, a term representing the sale of the changeover products can be added to the objective function. Then the objective function changes to the profit in a transition period, which is the sale minus the cost. In this way, the valuable changeover products are taken into account using the proposed method.

The scheduling problem can be further simplified by eliminating some decision variables. By substituting the production amount (6) into the demand equality (7), the variable w_i can be eliminated and the production time is given by

$$\Theta_i = \frac{D_i}{G_i} t_c, \ \forall i . \tag{16}$$

The equality (16) implies the production time of a product is a fixed percentage of the total cycle time. Substituting (16) into the sum of the production cost (13) yields the following equations:

$$\frac{1}{t_c}\sum_{k=1}^{N} \delta_k^p = C^r \sum_{i=1}^{N} \frac{\Theta_i}{t_c} F_i = C^r \sum_{i=1}^{N} \frac{D_i}{G_i} F_i . \tag{17}$$

It is seen that the total production cost is actually a constant and it can be dropped from the objective function. So the cost function becomes

$$\varphi_2 = \frac{1}{t_c}\sum_{k=1}^{N} \delta_k^t . \tag{18}$$

The inventory cost is simplified by substituting w_i in eq. (7) and Θ_i in eq. (16) into eq. (2), i.e.

$$\begin{aligned} \varphi_1 &= \frac{1}{t_c}\sum_{i=1}^{N} \frac{1}{2} C_i^s \left(G_i t_c - D_i t_c\right) \frac{D_i}{G_i} t_c \\ &= t_c \sum_{i=1}^{N} \frac{C_i^s D_i \left(G_i - D_i\right)}{2G_i} \end{aligned} \tag{19}$$

which is a linear function of the cycle time.

Since the sum of production time over products is equal to that over time slots, the sum of production time or the sum of the transition time is also a fixed percentage of the total cycle time

$$\sum_{k=1}^{N} \theta_k^p = \sum_{i=1}^{N} \Theta_i = \sum_{i=1}^{N} \frac{D_i}{G_i} t_c = \left(1-B\right) t_c , \tag{20}$$

$$\sum_{k=1}^{N} \theta_k^t = t_c - \sum_{k=1}^{N} \theta_k^p = B t_c , \tag{21}$$

where the coefficient is

$$B = 1 - \sum_{i=1}^{N} \frac{D_i}{G_i} . \tag{22}$$

Through substituting the cycle time by the sum of the transition times according to (21), the objective function becomes

$$
\begin{aligned}
\varphi &= A\sum_{k=1}^{N}\theta_k^t + B\frac{\sum_{k=1}^{N}\delta_k^t}{\sum_{k=1}^{N}\theta_k^t} \\
&= \frac{A\left(\sum_{k=1}^{N}\theta_k^t\right)^2 + B\sum_{k=1}^{N}\delta_k^t}{\sum_{k=1}^{N}\theta_k^t}
\end{aligned}
\tag{23}
$$

where the coefficient *B* is define in eq. (22) and *A* is

$$
A = \frac{1}{B}\sum_{i=1}^{N}\frac{1}{2}C_i^s\frac{D_i\left(G_i - D_i\right)}{G_i}. \tag{24}
$$

2.2 PI control problem

Information about the transition period, i.e. the transition time and the transition cost, is required in the scheduling problem. To retrieve the information, the control system design for the CSTR is taken into account. A common assumption in integration of scheduling and control is that the material flow rate $u(t)$ in the scheduling problem is a control variable in the control system (Flores-Tlacuahuac & Grossmann, 2006; Nystrom et al., 2005).

Though the trajectory of $u(t)$ can be calculated directly by the integrated problem (Flores-Tlacuahuac & Grossmann, 2006; Nystrom et al., 2005), an indirect calculation method is adopted in this work. The controller is assumed to have a parameterized structure and the controller parameters are calculated in the integrated problem. The trajectory of $u(t)$ is then determined based on the controller parameters as well as the measurements. The advantage of the indirect method is that even if the controller parameters are not updated by the integrated problem, the control system still works in the closed loop. In contrast, if the control variable is calculated directly by the integrated problem, the control system takes the risk of working in the open loop when the integrated problem cannot be solved between the two adjacent sampling points.

The most widely-used controller in the process industry is the PI controller (Ang et al., 2005), which is considered in this work. In the PI controller, the control variable $u(t)$ is

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)\, d\tau \,, \tag{25}$$

where $e(t)$ is the control error, which is the discrepancy between the measured output and the setpoint. The first term in the sum (25) is the proportional term while the second term is the integral term. Due to the integral term there is no steady state error in the control loop. The controller parameters are the proportional constant K_P and the integral constant K_I so the design of a PI controller is to tune the two parameters. There are various tuning rules, model free or model based (Ang et al., 2005; Skogestad, 2003). Optimization is a major tuning method in the PI controller design (Astrom et al., 1998; Kim et al., 2008; Toscano, 2005) that is able to take advantage of the mathematical model and the power of the modern optimization tools.

The optimization-based design method uses the dynamic model of the process and the model is frequently expressed by a set of differential and algebraic equations. Including the controller, the model of the closed-loop control system is

$$\dot{x}(t) = f\left(x(t), u(t)\right), \tag{26}$$

$$y(t) = g\left(x(t), u(t)\right), \tag{27}$$

$$e(t) = y_{sp} - y(t), \tag{28}$$

$$\dot{d}(t) = e(t), \tag{29}$$

$$u(t) = K_P e(t) + K_I d(t), \tag{30}$$

$$\dot{v}(t) = u(t). \tag{31}$$

In the state equations (26), all state variables are stacked into $x(t)$. The variable $u(t)$ is the control variable (or the process input) and $y(t)$ is the process output.

Controller design aims to let the output $y(t)$ track the given setpoint y_{sp} and the discrepancy in eq. (28) is defined as the control error $e(t)$. The control variable for the PI controller is driven by the proportional term and the integral term of the control error (30). To calculate the integral term, a differential equation (29) is introduced. Similarly, the differential equation (31) is introduced to calculate the integral of the control variable, which is equivalent to

$$v(t) = \int_0^t u(\tau)\, d\tau \text{ with } v(0) = 0 \,, \tag{32}$$

The control variable is also the material flow rate so $v(t)$ is the integrated material rate in the interval of [0, t].

The differential and algebraic equations of the closed-loop system can be expressed in a more compact form as

$$x^c(t) = f_c\left(x^c(t), z(t), K_P, K_I, y_{sp}\right) \text{ with } x^c(0) = x_0^c,$$
$$z(t) = g_c\left(x^c(t), z(t), K_P, K_I, y_{sp}\right), \tag{33}$$

where $x^c(t) = \left[x(t),\ d(t),\ v(t)\right]$ and $z(t) = \left[u(t),\ y(t),\ e(t)\right]$. The dynamic behavior of the closed-loop control system is dependent on three types of parameters: the initial value of x_0^c, the setpoint y_{sp}, and the controller parameters K_P and K_I. The initial value and the setpoint are determined by the scheduling problem while the controller parameters are the decision variables in the dynamic optimization problem.

Dynamic optimization with constraints of differential equations is an infinite-dimensional optimization problem. Discretization methods are typically applied to approximate the infinite-dimensional problem by a finite-dimensional problem. A common one is the simultaneous approach (Biegler, 2007; Cuthrell & Biegler, 1987), which discretizes all continuous time functions including the input, output, and states.

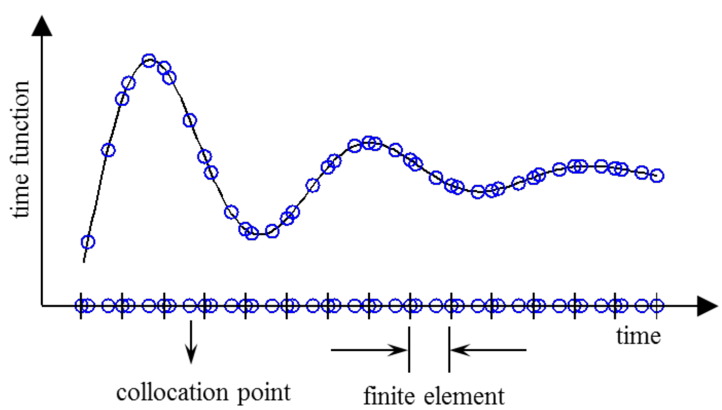


Fig. 2. Discretization of the continuous time function by collocation method.

An illustration of the discretization of a continuous time function is displayed in Fig. 2. First, a set of finite elements is used to partition the time interval into a number of equal-length segments. Then in each finite element, multiple collocation points are selected. Next, the continuous function is sampled at each collocation points. By approximating the continuous function with the discrete sampling points, the differential equations of the closed-loop control system can be approximated by a set of nonlinear algebraic equations as

$$
\begin{aligned}
x_{mp}^{c} &= x_{m,0}^{c} + h\sum_{r=1}^{N_{cp}} C_{rp}\dot{x}_{mp}^{c},\ x_{1,0}^{c} = x_{0}^{c}, \\
\dot{x}_{mp}^{c} &= f_{c}\left(x_{mp}^{c}, z_{mp}, K_{P}, K_{I}, y_{sp}\right), \\
z_{mp} &= g_{c}\left(x_{mp}^{c}, z_{mp}, K_{P}, K_{I}, y_{sp}\right),
\end{aligned}
\tag{34}
$$

where the variable x_{mp}^{c} is the discretized state $x^{c}(t)$ at the p-th collocation point in the m-th finite element and $x_{m,0}^{c}$ is the value at the starting collocation point in the m-th finite element, which is equal to the value at the ending collocation point in the previous finite element. The initial value of the state is determined by x_{0}^{c}. The variable of $z(t)$ is discretized into z_{mp} in the same way. The value of h denotes the length of the finite element and N_{cp} is the number of collocation points in each finite element. The coefficients C_{rp} are constants which are determined by the specific collocation approach used for the discretization. After discretization, the differential and algebraic equations (33) are approximated by a set of purely algebraic equations (34) which can be added as constraints into the integrated optimization problem.

2.3 Simultaneous scheduling and control problem

After discretization, a nonlinear optimization solver can be applied to solve the control problem with constraints (34). In the integrated problem, the control problem is solved for each time slot. On one hand, the controller design requires information from the scheduling decisions. The initial value and the setpoint for the control problem in slot k is provided by the scheduling problem as

$$
x_{0}^{c,k} = \sum_{i=1}^{N} \xi_{ik} E_{i},\ \forall k \tag{35}
$$

$$
y_{sp}^{k} = \begin{cases} \sum_{i=1}^{N} \xi_{ik+1} H_{i},\ k \neq N \\ \sum_{i=1}^{N} \xi_{i1} H_{i}, \quad k = N \end{cases} \tag{36}
$$

The initial value is obtained from the data of the production period in the current slot while the setpoint is from the information of the production period in the next slot. The constants E_{i} and

H_i are the steady-state values for manufacturing product i and they are available from the production recipe.

On the other hand, the control problem provides the transition cost and the transition time for the scheduling problem. The transition time is defined as the duration in which the process transfers from one production period to another production period. To define the production period, a quality bound around the steady-state value of the output is introduced. The production period ranges from the time at which the output falls into the bound to the time when the next changeover begins. During the production period, the output needs to continuously stay between the bounds. The transition time is determined as the time at which the output enters the quality bound. Mathematically, the transition time can be determined by the constraints

$$\begin{aligned} &\theta_k^t = M_t h_k \\ &y_{mp}^k \le y_{sp}^k (1+b), \quad \forall m > M_t \\ &y_{mp}^k \ge y_{sp}^k (1-b), \quad \forall m > M_t \end{aligned} \tag{37}$$

where M_t is the index of the finite element after which the output stays between the bounds. The width of the bound is defined by the value of b. In the optimization, M_t is set as a constant while the length of the finite elements h_k is a variable. The value of b is typically chosen as 2%. The material cost in the transition period, or the transition cost in brief, is the integral of the flow rate and it is equal to the variable $v^k(t)$ in slot k evaluated at the transition time of θ_k^t, i.e.

$$\delta_k^t = C^r v^k \left(\theta_k^t\right) = C^r v_{M_t, N_{cp}}^k , \tag{38}$$

where the transition cost is equal to the discretized v^k at the last collocation point in the finite element indexed by M_t.

Integrated problem

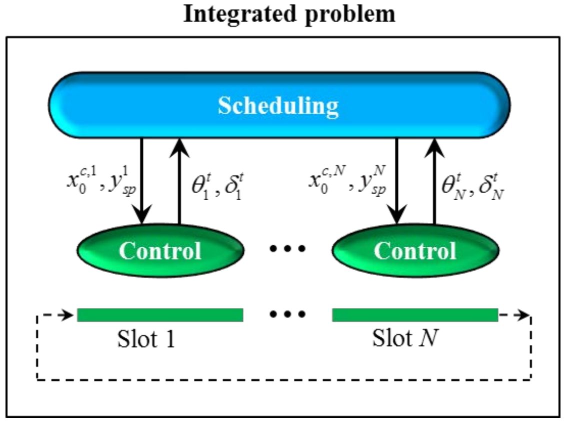


Fig. 3. The structure of the integrated problem.

To manifest the coupling between the scheduling problem and the control problems, the structure of the integrated MIDO is shown in Fig. 3. The scheduling problem in the upper level interacts with multiple control problems in the lower level. The number of the control problems is equal to the number of the time slots. The scheduling problem requires information of the transition time and the transition cost from all control problems while a control problem in each slot needs the initial condition and the setpoint provided by the scheduling problem. The model of the integrated problem is summarized as

$$
\begin{array}{lll}
\min \ \varphi & (23) & \\
\text{s.t.} & & \\
\quad \text{Product and slot assignment} & (4), (5) & \\
\quad \text{Control system model in each slot} & (34) & \\
\quad \text{Initial condition and setpoint} & (35), (36) & \\
\quad \text{Transiton time and transition cost} & (37), (38) &
\end{array}
\tag{39}
$$

The decision variables can be cast into 4 groups

$$
\begin{array}{l}
\left\{\xi_{ik}\right\} \text{ collection of assignment variables} \\
\left\{\theta_k^t\right\} \text{ collection of transition times} \\
\left\{\delta_k^t\right\} \text{ collection of transition costs} \\
\left\{\eta_k\right\} \text{ collection of variables in control problems}
\end{array}
\tag{40}
$$

The variable η_k is introduced to represent the variables in the control problem in slot *k*, including the controller parameters and all discretized input, output and state variables.

The integrated problem (39) after discretization of the dynamic model is an MINLP. A straightforward approach is to solve it directly by an MINLP solver and this is the simultaneous approach (Flores-Tlacuahuac & Grossmann, 2006, 2011). However, the MINLP problem could have a very large scale because multiple control sub-problems are required to solve simultaneously. Since the production sequence is not known in advance, we do not know the specific dynamic control problem in each time slot. Therefore, it is difficult to determine the number of finite elements for the dynamic optimization problem in a given time slot. The number of finite elements is critical in the discretization method. If it is too large, the number of equations and variables increase and the discretized model requires a great amount of computational resources. However, the number cannot be too small either. Otherwise, there will be a significant discretization error, or even the occurrence of infeasible solutions. When the dynamic optimization problem is solved individually, the number of finite elements can be set according to the given transition. In the integrated problem, all transitions are possible so the number of the finite elements have to be set large enough to meet the worst case even though it is redundant for other cases. The complexity in the formulated model incurs the difficulties in the solution feasibility and computational efficiency.

3. Solution strategy of the integrated problem

Though the integrated problem formulated in the previous section can be solved directly by an MINLP solver, this approach might be computationally very expensive. These difficulties become more severe for online application where a more stringent requirement is posed on the reliability and the efficiency. Instead of simply leaving all complexities to the solver, it is more helpful to simplify the solution procedure by exploring the model structure for such a difficult MIDO problem (Allgor & Barton, 1999).

In this section, the model structure is first analyzed and a decomposition method which separates the dynamic optimization problems from the integrated problem is proposed in Section 3.1. After the decomposition, the integrated problem is approximated by a simultaneous scheduling and controller selection problem. The resulting model is computationally much more efficient than the original integrated problem and it has a special structure which can be further exploited to design an efficient global optimization algorithm. The solution procedure is presented in Section 3.2.

3.1 Decomposition of the integrated problem

Solving the integrated problem might be a challenging task, however, the individual scheduling problem or a single control problem in one transition period is easy to solve. The difficulties only stem from the combination where the control problems for all transition periods are solved simultaneously with the scheduling decisions. Therefore, if the two types of problems can be solved separately, the integrated problem will be easily solved. To achieve this goal, we develop a decomposition method for the MIDO problem. The decomposition is conducted based on the bilevel optimization procedure and the approximation of the optimal-value function for the inner optimization problem.

For the bilevel programming, the integrated problem is divided into an inner optimization problem and an outer optimization problem

$$\begin{aligned}
&\min_{\{\xi_{ik}\},\{\theta_k^t\},\{\delta_k^t\},\{\eta_k\}} \varphi \\
\Leftrightarrow &\min_{\{\xi_{ik}\},\{\theta_k^t\}} \left\{ \min_{\{\delta_k^t\},\{\eta_k\}} \left\{ \frac{A\left(\sum_{k=1}^{N}\theta_k^t\right)^2 + B\sum_{k=1}^{N}\delta_k^t}{\sum_{k=1}^{N}\theta_k^t} \right\} \right\}
\end{aligned} \tag{41}$$

$$\Leftrightarrow \min_{\{\xi_{ik}\},\{\theta_k^t\}} \left\{ \frac{A\left(\sum_{k=1}^{N}\theta_k^t\right)^2 + B\min_{\{\delta_k^t\},\{\eta_k\}}\left\{\sum_{k=1}^{N}\delta_k^t\right\}}{\sum_{k=1}^{N}\theta_k^t} \right\} \tag{42}$$

According to the groups of the decision variables in the integrated problem (40), the inner optimization problem is solved with the decision variables $\{\delta_k^t\}$ and $\{\eta_k\}$, while the outer optimization is solved with the decision variables $\{\xi_{ik}\}$ and $\{\theta_k^t\}$. In the inner optimization problem, the decision variables $\{\xi_{ik}\}$ and $\{\theta_k^t\}$ in the outer optimization problem can be treated as parameters. So the inner optimization can be reformulated as the minimization of the sum of transition costs in the numerator (42).

The inner optimization problem can be further simplified according to the structure of the integrated problem. The derivation procedure is listed below and detailed explanations are given in the following paragraphs.

$$\min\sum_{k=1}^{N}\delta_k^t \tag{43}$$

$$\Leftrightarrow \sum_{k=1}^{N}\min\delta_k^t \tag{44}$$

$$\Leftrightarrow \sum_{k=1}^{N}\delta_k^{t*}\left(\{\xi_{ik}\},\{\theta_k^t\}\right) \tag{45}$$

$$\Leftrightarrow \sum_{k=1}^{N}\delta_k^{t*}\left(\xi_{ik},\xi_{jk+1},\theta_k^t\right) \tag{46}$$

$$\Leftrightarrow \sum_{k=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_{ijk}\delta_{ij}^{t*}\left(\theta_{ij}^t\right) \tag{47}$$

An important observation from the structure of the integrated problem (Fig. 3) is that when the assignment decisions $\{\xi_{ik}\}$ are given, the transition sequence is known and the dynamic optimization in a time slot can be solved independently from the dynamic optimization in the other slots. Therefore, the optimization on the sum of transition costs is equivalent to the sum of the optimization on each term (44).

The dynamic optimization which minimizes the transition cost in each time slot can be solved separately. The optimal solution to the inner optimization problem in slot k turns out to be

$$\delta_k^{t*}\left(\{\xi_{ik}\}\ ,\{\theta_k^t\}\right) = \min \delta_k^t . \tag{48}$$

The sum of the optimal-value functions (45) is obtained. The optimal-value function characterizes the dependence of the optimal value on the parameters in the optimization problem. Each optimal-value function $\delta_k^{t*}\left(\{\xi_{ik}\}\ ,\{\theta_k^t\}\right)$ can be further simplified. Because the assignment variables are used to calculate the initial condition (35) and the setpoint (36) for the control system, the transition cost in slot k is only dependent on the assignment variables ξ_{ik} and ξ_{ik+1}. Similarly, only the transition time in the current slot θ_k^t has effects on the transition cost since it adds constraint (37) to the dynamic optimization. The optimal-value function is only dependent on the three variables as shown in the equality (46).

Furthermore, the dependence of the dynamic optimization on the time slot can be removed because the initial condition and the setpoint will be known directly if the transition sequence $i \rightarrow j$ is given. To model the transition sequence, a set of 0-1 variables β_{ijk} is introduced. The binary variable β_{ijk} denotes if product i in slot k is followed by product j. Then the optimal-value function is only dependent on the transition time θ_{ij}^t in the transition from product i to product j. By substituting the optimal-value function, the expression (47) is obtained. The optimal transition cost is calculated from

$$\delta_{ij}^{t*}\left(\theta_{ij}^t\right) = \min\ \delta_{ij}^t \tag{49}$$

s.t.

$$\begin{aligned}
&\text{Control system model } (34)\\
&x_0^c = E_i\\
&y_{sp} = H_j\\
&h = \theta_{ij}^t / M_t
\end{aligned}$$

The initial condition and the setpoint are set directly from the steady-state data and the length of the finite element, *h*, is determined by the value of the transition time θ_{ij}^{t}.

The optimal-value function $\delta_{ij}^{t*}\left(\theta_{ij}^{t}\right)$ is now only dependent on the single transition time and it can be substituted back to the outer optimization problem (42). Then the integrated problem is solved with only the decision variables $\left\{\xi_{ik}\right\}$ and $\left\{\theta_{k}^{t}\right\}$. The constraints and variables of the control problems are all encapsulated inside the inner optimization problem (49). Therefore, the outer optimization problem is much simpler than the original integrated problem. However, the analytical expression of $\delta_{ij}^{t*}\left(\theta_{ij}^{t}\right)$ is not known and the direct substitution is not applicable. An approximation of the optimal-value function is obtained via discretization.

The feasible range of the transition time θ_{ij}^{t} is first discretized into a set of sampling points. Then for each sampled transition time the dynamic optimization is solved and the optimal solution of the transition cost is recorded. Finally the function $\delta_{ij}^{t*}\left(\theta_{ij}^{t}\right)$ is approximated by a set of sampling points and the discrete points are substituted back to the outer optimization problem instead of the optimal-value function.

To determine the feasible range of the transition time, a dynamic optimization problem is solved which minimizes the transition time

$$\theta_{ij,\min}^{t} = \min \theta_{ij}^{t} . \tag{50}$$

The minimum transition time $\theta_{ij,\min}^{t}$ is the lower bound of the feasible range and serves as the starting point of the discretization. The discretization resolution is determined by the length between the sampling points, which is set by an incremental factor D_{θ}. Then the range of the transition time is discretized by $\theta_{ijl}^{t} = \theta_{ij,\min}^{t} + \left(l-1\right)\cdot D_{\theta}$, where l is a positive integer. The minimum transition time is the one corresponding to l = 1. Though the transition time can be set to any large value by choosing the controller parameters in the control system, it is often bounded from above in the scheduling problem. So an upper bound of N_C is set on the index of the sampling points, i.e. $l \in \left\{1, 2, \cdots, N_C\right\}$. Selection of N_C can be validated from the objective function value of the integrated problem and the detailed procedure will be discussed in the section of Case Studies. For each sampled transition time, we solve a dynamic optimization problem to minimize the transition cost

$$\begin{aligned} \delta_{ijl}^t &= \min \ \delta_{ij}^t \\ \text{s.t.} \ \ \theta_{ijl}^t &= \theta_{ij,\min}^t + (l-1)\cdot D_\theta \end{aligned} \tag{51}$$

where the optimal value of the transition cost is indexed by the number of the sampling point, l. In this way, the optimal-value function $\delta_{ij}^{t*}\left(\theta_{ij}^t\right)$ is approximated by a set of discrete values $\left(\theta_{ijl}^t,\ \delta_{ijl}^t\right)$, $1 \le l \le N_C$.

The set of discrete values from the optimal-value function is substituted into the outer optimization problem. Then one from the set will be selected. To model the selection, a set of binary variables γ_{ijl} is introduced. The transition time and the transition cost are expressed as

$$\begin{aligned} \theta_{ij}^t &= \sum_{l=1}^{N_C} \gamma_{ijl}\theta_{ijl}^t \\ \delta_{ij}^t &= \sum_{l=1}^{N_C} \gamma_{ijl}\delta_{ijl}^t. \end{aligned} \tag{52}$$

The transition time and the transition cost are essentially determined by the controller returned by the dynamic optimization. Therefore the selection of the transition time and the transition cost means the associated controller is selected in the integrated problem, i.e. if the l-th controller in the transition from product i to product j is selected then $\gamma_{ijl} = 1$.

It should be noted that controller design is dependent on the transition sequence of $i \rightarrow j$. For different transitions, the calculated controller parameters are generally different. The dependence of the controller parameters on the transition sequence is a main reason that the controller design and the scheduling problem should be solved in an integrated way.

The transition time in a time slot is expressed as a combination of θ_{ij}^t

$$\theta_k^t = \sum_{i=1}^{N}\sum_{j=1}^{N} \beta_{ijk}\theta_{ij}^t = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{N_C} \beta_{ijk}\gamma_{ijl}\theta_{ijl}^t, \tag{53}$$

where binary variables β_{ijk} are introduced to denote if product j follows product i in slot k. The objective function of the integrated problem is dependent on the sum of the transition time over all time slots, which is defined as the total transition time, t_T

$$t_T = \sum_{k=1}^{N} \theta_k^t = \sum_{k=1}^{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{l=1}^{N_C} \beta_{ijk} \gamma_{ijl} \theta_{ijl}^t \ . \tag{54}$$

Similarly the total transition cost, denoted by c_T, is the sum of all transition costs

$$c_T = \sum_{k=1}^{N} \delta_k^t = \sum_{k=1}^{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{l=1}^{N_C} \beta_{ijk} \gamma_{ijl} \delta_{ijl}^t \ . \tag{55}$$

By substituting the total transition time and the total transition cost, the integrated problem is transformed into a simultaneous scheduling and controller selection problem, which is formulated as

$$\min \ \varphi = \frac{At_T^2 + Bc_T}{t_T} \tag{56}$$

s.t.

Assignment constraints $(4), (5)$

$$\sum_{j=1}^{N} \beta_{ijk} = \xi_{ik}, \ \forall i, k \tag{57}$$

$$\sum_{i=1}^{N} \beta_{ijk} = \begin{cases} \xi_{jk+1} & k \neq N \\ \xi_{j1} & k = N \end{cases}, \ \forall j, k \tag{58}$$

$$\sum_{l=1}^{Nc} \gamma_{ijl} = \sum_{k=1}^{N} \beta_{ijk}, \ \forall i, j \tag{59}$$

$$t_T = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{l=1}^{Nc} \gamma_{ijl} \theta_{ijl}^t \tag{60}$$

$$c_T = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{l=1}^{Nc} \gamma_{ijl} \delta_{ijl}^t \tag{61}$$

The assignment constraints come from the original scheduling problem. The sequence variable β_{ijk} is related to the assignment variable ξ_{ik} according to the constraints (57) and (58). Since the products are manufactured cyclically, the last product in the current cycle is followed by the first product in the next cycle. The constraints suggest the binary sequence variables, β_{ijk}, can be replaced by continuous variables between 0 and 1 (Wolsey, 1997). This significantly reduces the number of discrete variables and improves the computational efficiency.

The controller selection is required only when the transition *ij* appears in the production sequence. To check it, the sequence variables β_{ijk} is summed up over the slot *k*. The sum is equal to one if the transition appears in the production sequence and it is equal to zero, otherwise. The sum of the selection variables γ_{ijl} over the controller candidate l is set to the sum of the sequence variables (59). When the transition ij appears in the production sequence, one controller is selected from the candidates. Otherwise, no selection is required and all γ_{ijl} for the given ij can be equal to zero. Then the total transition time (54), which includes the bilinear term $\beta_{ijk}\gamma_{ijl}$, is simplified into the constraint (60), i.e.

$$\begin{aligned} t_T &= \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{N_C}\left(\sum_{k=1}^{N}\beta_{ijk}\right)\gamma_{ijl}\theta_{ijl}^{t} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{N_C}\sum_{l'=1}^{N_C}\gamma_{ijl'}\gamma_{ijl}\theta_{ijl}^{t} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{N_C}\gamma_{ijl}\theta_{ijl}^{t}. \end{aligned} \tag{62}$$

The last equality is derived from the fact that $\gamma_{ijl'}\gamma_{ijl} = 0$ for $\forall l' \neq l$ and $\gamma_{ijl'}\gamma_{ijl} = \gamma_{ijl}$ for $l' = l$ because at most one selection variable can be 1. In the same way, the total transition cost (55) is simplified as the constraint (61). All constraints are linear, making the problem easier to solve.

The summary of the decomposition method for solving the integrated problem is displayed in Fig. 4. The dynamic optimization problem is solved at each discretized transition point for each transition and there are $N\times(N-1)\times N_C$ dynamic optimization problems in total. However, all these problems can be solved offline and each of them is a standard dynamic optimization problem without integer variables that can be solved easily. The integrated problem is approximated by the problem of simultaneous scheduling and controller selection and the complexity is significantly reduced, because all variables and equations in the control problems are removed. The online implementation only includes the simultaneous scheduling and controller selection problem which can be solved to global optimality efficiently.

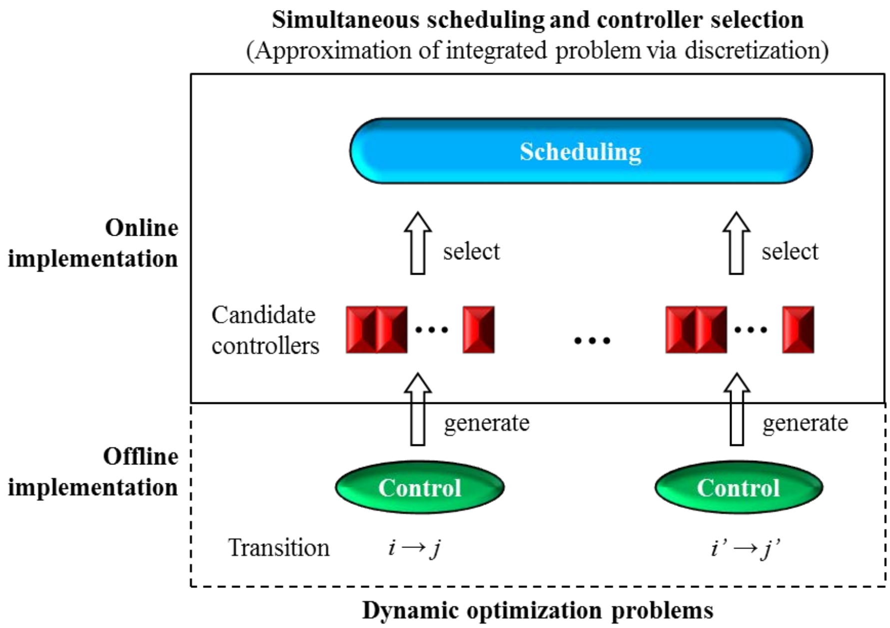


Fig. 4. Diagram of the decomposition method for the integrated problem.

The controller design procedure in the integration method aims to improve the economic objective where conventional controller design methods frequently pay attention to stability and robustness issues. Though the two issues are not the focus of the paper, it will be helpful to make a brief discussion about stability and robustness of the integration method. In fact the proposed integration method also has some advantages on the two issues comparing with the simultaneous approach.

In the proposed integration method, the PI controllers are designed offline and the design in a transition is conducted independently from that in another transition. Stability of the closed-loop control system in a transition period can be analyzed in a traditional way (Soylemez et al., 2003; Tan et al., 2006) which is applied to a single control system. In practice, it is easy to detect unstable controllers by simulating the closed-loop system. The unstable controllers will be eliminated and thus the controller candidates for the integrated problem are all stable. In the integration method, no matter which controllers are selected, the selected controllers can guarantee stability of the closed-loop system in each transition period. Further, because any two transition periods are separated by a production period where the process stays at the steady state, stability of a transition period dose not interact with that in another period. Therefore, stability of the control system in each transition period implies the stability of the integration method in a whole cycle. In contrast, the simultaneous approach determines trajectories of control variables in all transition periods simultaneously with scheduling decisions. Because the transition sequences as well as the controllers are not known a priori, analysis of stability for the integrated model is more difficult.

Similarly to stability analysis, robustness can be investigated on a single control system in a transition period. Various robust design approaches for the PI controller have been developed (Huang & Wang, 2000; Toscano, 2005) where process uncertainties are taken into account explicitly in the design procedure. However, considering the integrated problem has already been very complicated, process uncertainties are not modeled directly in the design procedure of the integration method. Robustness of the integration method is achieved by rescheduling the process, which will be illustrated in the case study. At the beginning of a production period, the information of the production process is updated after the occurrence of uncertainties. Then the schedule and the controller parameters are recalculated according to the updated information. Uncertainties are handled in a reactive way instead of the proactive way (Huang & Wang, 2000; Toscano, 2005) which is common in traditional robust controller design procedure. In comparison, due to the computational complexity, the simultaneous approach is difficult to implement online to deal with uncertainties.

3.2 Fast solution strategy for the simultaneous scheduling and controller selection problem

The simultaneous scheduling and controller selection problem includes much fewer constraints and variables than the original integrated problem. The objective function (56) is a fraction with a quadratic function in the numerator and a linear function in the denumerator. All constrains are linear. The NLP relaxation of this problem can be considered as a fractional programming problem with special structure.

A general expression of the fractional programming problem is given by

$$\begin{aligned} \min\ & Q(x) = N(x)/D(x) \\ \text{s.t.}\ & x \in S \end{aligned} \tag{PI}$$

where $N(x)$ and $D(x)$ are functions in the numerator and the denominator respectively, S denotes the feasible range which is assumed to be a convex set. The value of the denominator is assumed to be positive, $D(x) > 0$. A nonlinear fractional function might not be convex, however, the quasi-convexity can be obtained in a case of

$$\begin{aligned} \min \ & Q(x) = \frac{x^{\mathrm{T}} P x + a^{\mathrm{T}} x + a_0}{b^{\mathrm{T}} x + b_0} \\ \text{s.t.} \ & x \in S \end{aligned} \tag{PII}$$

where P is a symmetric matrix, a and b are vectors, a_0 and b_0 are scalars. This is a special fractional function with a quadratic function in the numerator while an affine function in the denominator.

Lemma 1 (Quasi-convexity of the quadratic fractional programming). (Cambini et al., 2002) The problem P(II) is quasi-convex if $v^{\mathrm{T}} P v \geq 0$ for $\forall x, v$ such that $\nabla Q(x)^{\mathrm{T}} v = 0$.

If the simultaneous scheduling and controller selection problem (56) is expressed as the general form (PII), the quadratic terms stem only from the square of the total transition time and they can be expressed in the compact matrix form as

$$x^{\mathrm{T}} P x = A t_T^2 . \tag{63}$$

where x contains all decision variables. According to the equality, for any x, the value of the quadratic term is always non-negative due to the square operation on the right hand side and the positive value of the parameter A. Therefore, the matrix P is positive semi-definite and the condition in Lemma 1 is satisfied. In fact the condition of the positive semi-definite P is stronger than the condition in Lemma 1 and it guarantees that the quadratic term of $v^{\mathrm{T}} P v$ is always non-negative for all v while the condition in Lemma 1 only requires the non-negativity for the vectors orthogonal to the gradient of the objective function. Under the condition of the positive semi-definite P, there is no need to check the vectors a and b, or the scalars a_0 and b_0 in the objective function. The conclusion is summarized in Proposition 1.

Proposition 1.

The NLP relaxation of simultaneous scheduling and controller selection problem is quasi-convex.

The property of being quasi-convex implies that an NLP solver can globally optimize the relaxed problem of scheduling and controller selection. The MINLP solver SBB (Bussieck & Drud, 2001) has the advantage for solving such kind of problems because each NLP subproblem in the branch-and-bound is solved to the global optimum so the global solution to the MINLP problem can be obtained by a standard branch-and-bound algorithm. However, another popular MINLP solver DICOPT (Kocis & Grossmann, 1989) uses the outer-approximation method and it might cut off some feasible solution space during the iterations. Thus, DICOPT cannot guarantee the global optimality of the solution. The performance of these algorithms will be illustrated in the subsequent case studies.

The global MINLP solver BARON (Tawarmalani & Sahinidis, 2004) can also be used to calculate the global optimal solution of the simultaneous scheduling and controller selection problem. However, a more efficient approach for its global optimization is the method via parametric programming. The fractional programming problem (PI) can be transformed into a parametric programming problem (PIII) (Bradley & Arntzen, 1999; Dinkelbach, 1967; You et al., 2009)

$$\begin{aligned} F(q) = \min \quad & \{N(x) - qD(x)\} \\ \text{s.t.} \quad & x \in S \end{aligned} \tag{PIII}$$

where q is a parameter and the optimal value is a function of the parameter, $F(q)$. The two problems are equivalent according to Lemma 2.

Lemma 2 (Equivalence of the fractional programming problem and the parametric programming problem). (You et al., 2009)
The variable x^* is the global solution to the fractional programming problem (PI) if and only if x^* is the global solution to the parametric programming problem (PIII) with the parameter q^* such that $F(q^*) = 0$.

It should be noted that the condition for the equivalence lemma is very general and only the positive denominator is required, i.e. $D(x) > 0$. The problems can include both continuous and discrete variables, and the functions $N(x)$ and $D(x)$ can be general. The parametric programming problem (PIII) is generally simpler than the fractional programming problem (PI) since the ratio is transformed into the difference. The global solution to the parametric programming problem is also the global solution to the fractional programming problem according to Lemma 2. Though there are no assumptions on the convexity of the two functions $N(x)$ and $D(x)$, the equivalence is only valid based on the global solutions of both problems, not on the local solutions. Thus $N(x)$ is often assumed to be convex and $D(x)$ is assumed to be concave so that the parametric programming is convex with a positive value of q where a local solution is also the global solution. In the equivalence proposition, the zero of the optimal-value function $F(q)$ needs to be calculated and the calculation is based on the properties of the function.

Lemma 3 (Properties of the optimal-value function). (You et al., 2009)

(a) $F(q)$ is concave

(b) $F(q)$ is continuous

(a) $F(q)$ is strictly decreasing

(d) $F(q) = 0$ has a unique solution

Due to the properties of $F(q)$ listed in Lemma 3, the equation $F(q) = 0$ has a unique solution and it can be calculated by a regular nonlinear equation solver. The property of being strictly decreasing motivates the bisection method (Bradley & Arntzen, 1999). Assume the solution is q^* which has $F(q^*) = 0$. For a given value of q, if the objective function value of $F(q)$ is positive

$$F(q) > 0 = F(q^*) \tag{64}$$

then the solution q^* must be larger than q given that the function is strictly decreasing. Thus, the range with the value less than q can be safely excluded from the following search and the lower bound of the updated search range can be set as q. Similarly if the function value of $F(q)$ is negative

$$F(q) < 0 = F(q^*) \tag{65}$$

the range with the value larger than q can be excluded and the upper bound can be set as q. If the new value of q is selected as the middle point between the lower bound and the upper bound, the search range shrinks by a half after each iteration step. The bisection method will converge to the solution since the length of the search interval decreases to zero.

Another alternative to solve the equation $F(q) = 0$ is the Dinkelbach's method (You et al., 2009). The method is based on the generalized Newton's algorithm by using the subgradient. The subgradient of the optimal-value function can be calculated from the relations

$$
\begin{aligned}
& F(q_2) - F(q_1) \\
&= \min \ \{N(x) - q_2 D(x)\} - \min \ \{N(x) - q_1 D(x)\} \\
&= \min \ \{N(x) - q_2 D(x)\} - \{N(x_1^*) - q_1 D(x_1^*)\} \\
&\leq \{N(x_1^*) - q_2 D(x_1^*)\} - \{N(x_1^*) - q_1 D(x_1^*)\} \\
&= (q_2 - q_1)(-D(x_1^*))
\end{aligned}
\tag{66}
$$

where x_1^* is the optimal solution to the parametric programming

$$
x_1^* = \arg\min \ \{N(x) - q_1 D(x)\}. \tag{67}
$$

The solution of x_1^* is optimal when the parameter $q = q_1$, however, it is only suboptimal when $q = q_2$. The non-optimality causes the inequality in (66) and the subgradient of the optima-value function is identified as $-D(x_1^*)$, which is the negative value of the denominator evaluated at the optimal solution of x_1^*. The subgradient can be then utilized in the Newton's algorithm, resulting in the iteration

$$
\begin{aligned}
q_{n+1} &= q_n - \frac{F(q_n)}{-D(x_n^*)} \\
&= q_n + \frac{N(x_n^*) - q_n D(x_n^*)}{D(x_n^*)} \\
&= \frac{N(x_n^*)}{D(x_n^*)}
\end{aligned}
\tag{68}
$$

The updated parameter value at the next step is equal to the value of the fractional function in the current step.

The methods via parametric programming can be applied to the simultaneous scheduling and controller selection problem (56). From the expression (63), the matrix P is positive semi-definite so the quadratic function in the numerator is convex. The convexity is independent on the coefficients of linear terms. The denominator is a linear function, which is concave. So the parametric programming problem corresponding to the simultaneous scheduling and controller selection problem is convex. The property is summarized in Proposition 2.

Proposition 2.

The parametric programming problem corresponding to the simultaneous scheduling and controller selection problem, which is

$$F(q) = \min \ \left\{ At_T^2 + Bc_T - qt_T \right\}, \tag{69}$$

is convex for any q so the global solution to the problem can be obtained in an efficient way.

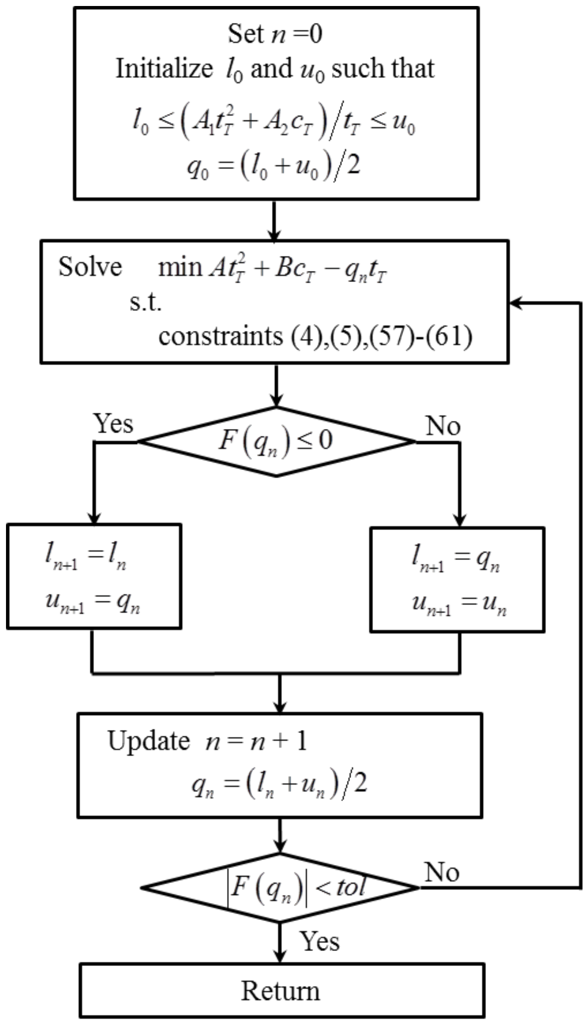


Fig. 5. Bisection method for solving the simultaneous scheduling and controller selection problem.

The parametric problem can be solved by either the bisection method or the Dinkelbach's method. Both methods solve the problem iteratively while the way in which the parameter q is updated is different. The procedure by the bisection method is shown in Fig. 5, where only the sign information of the function is used in the iterations. The procedure of the Dinkelbach's method is displayed in Fig. 6 and the method exploits the information of the entire function value. It will be seen in the case study that both methods via the parametric programming are much more efficient in finding the global optimal solution than the methods, which solve the mixed-integer nonlinear fractional problem directly. Comparing the two methods, the Dinkelbach's method is more efficient with fewer iteration steps and less computation time.

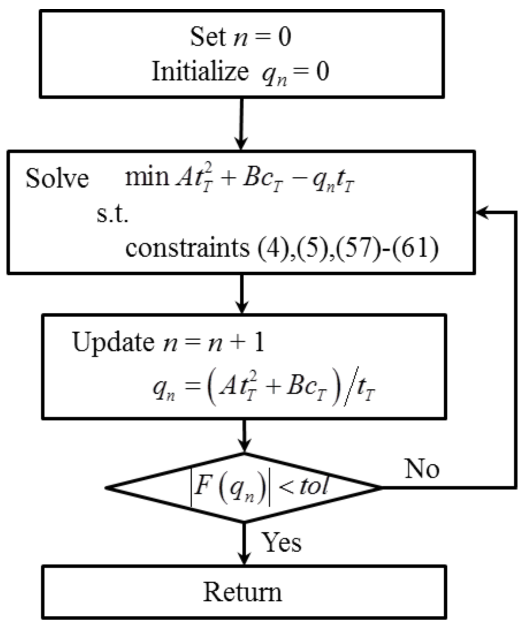


Fig. 6. Dinkelbach's method for solving the simultaneous scheduling and controller selection problem.

4. Case Study

The proposed integration method is applied to a methyl methacrylate (MMA) polymerization process. The dynamic model under consideration is a nonlinear free radical polymerization with azobisisobutyronitrile as initiator and toluene as the solvent. The set of differential equations (Mahadevan et al., 2002; Terrazas-Moreno et al., 2008) is summarized below and the parameters in the model are listed in Table 1.

$$
\begin{aligned}
\frac{dC_m}{dt} &= -\left(k_p + k_{f_m}\right)\sqrt{\frac{2f^*k_I}{k_{T_d}+k_{T_c}}}C_m\sqrt{C_I} + \frac{F\left(C_{m_{in}} - C_m\right)}{V} \\
\frac{dC_I}{dt} &= -k_I C_I + \frac{F_I C_{I_{in}} - FC_I}{V} \\
\frac{dD_0}{dt} &= \left(0.5k_{T_c} + k_{T_d}\right)\frac{2f^*k_I}{k_{T_d}+k_{T_c}}C_I + k_{f_m}\sqrt{\frac{2f^*k_I}{k_{T_d}+k_{T_c}}}C_m\sqrt{C_I} - \frac{FD_0}{V} \\
\frac{dD_1}{dt} &= M_m\left(k_p + k_{f_m}\right)\sqrt{\frac{2f^*k_I}{k_{T_d}+k_{T_c}}}C_m\sqrt{C_I} - \frac{FD_1}{V} \\
y &= \frac{D_1}{D_0} \\
u &= F_I
\end{aligned}
\tag{70}
$$

Table 1

Data of the MMA polymerization model

Symbol	Variables/Parameters	Value
C_m	Monomer concentration	kmol/m^3
C_I	Initiator concentration	kmol/m^3
D_0	Molar concentration of dead chains	kmol/m^3
D_1	Mass concentration of dead chains	kg/m^3
D_1/D_0	Molecular weight	kg/kmol
F_I	Initiator flow rate	m^3/h
F	Monomer flow rate	10.0 m^3/h
V	Reactor volume	10.0 m^3
f^*	Initiator efficiency	0.58
k_p	Propagation rate constant	2.50×10^6 m^3/kmol•h
k_{T_d}	Termination by disproportionation rate constant	1.09×10^{11} m^3/kmol•h
k_{T_c}	Termination by coupling rate constant	1.33×10^{10} m^3/kmol•h
$C_{I_{in}}$	Inlet initiator concentration	8.00 kmol/m^3
$C_{m_{in}}$	Inlet monomer concentration	6.00 kmol/m^3
k_{f_m}	Chain transfer to monomer rate constant	2.45×10^3 m^3/kmol•h
k_I	Initiation rate constant	1.02×10^{-1} h^{-1}
M_m	Molecular weight of monomer	100.12 kg/kmol

The state variables are the concentration of the monomer C_m, the concentration of the initiator C_I, the molar concentration of dead chains D_0, and the mass concentration of dead chains D_1. The output $y(t)$ is the molecular weight for each grade of the product and the manipulated input $u(t)$ is the flow rate of the initiator. The reaction takes place isothermally at the temperature of 335K. There are 16 products with different grades manufactured cyclically in the CSTR and each grade is designated by the value of the molecular weights.

The products are manufactured at the steady state and the steady-state values of the input, the output, and the state variables are listed in Table 2. The original model is poorly scaled. There are huge differences in the numerical values among the variables. For example, the steady-state values of the output variable are approximately 6 orders larger than those of the input variable. The poor scaling will cause severe difficulties for the subsequent optimization processes and therefore a scaling procedure is required to generate a well-scaled model. For this purpose, the mean value of a variable over the 16 products is calculated first and then each variable is divided by the mean value. After the scaling procedure, each variable becomes dimensionless and has the comparable numerical values. However, for easy understanding, the results presented in this section use the values with the original units, which are transformed back from the scaled values.

Table 2
Data for each product

Product	y_{ss} (10^4 kg/kmol)	u_{ss} (10^{-2} m^3/h)	$C_{m,ss}$ (kmol/m^3)	$C_{I,ss}$ (10^2 kmol/m^3)	$D_{0,ss}$ (10^4 kmol/m^3)	$D_{1,ss}$ (kg/m^3)
A	1.5	53.08	5.174	42.03	55.11	82.67
B	2.5	16.95	5.504	13.42	19.88	49.69
C	3.5	6.961	5.672	5.512	9.377	32.82
D	4.5	3.163	5.775	2.505	5.006	22.53
E	1.9	32.05	5.338	25.38	34.88	66.28
F	2.7	13.99	5.546	11.09	16.85	45.51
G	3.3	8.224	5.645	6.513	10.76	35.51
H	3.9	5.041	5.719	3.992	7.219	28.16
I	1.7	40.82	5.263	32.33	43.38	73.75
J	2.1	25.59	5.401	20.27	28.53	59.92
K	2.3	20.71	5.456	16.41	23.68	54.45
L	2.9	11.65	5.583	9.227	14.41	41.79
M	3.1	9.762	5.616	7.731	12.41	38.48
N	3.7	5.914	5.697	4.683	8.211	30.38
O	4.1	4.308	5.739	3.411	6.371	26.12
P	4.3	3.689	5.758	2.921	5.639	24.25

The CSTR operates with the constant volume so the outlet flow rate is equal to the inlet flow rate F, which is assumed to be constant for all products. The outlet flow rate determines the production rate and we have $G_i = F = 10$ m^3/h. The demand rates of all products are assumed to be equal as $D_i = 0.5$ m^3/h, the inventory cost rate is C_i^s = \$10/m^3•h, and the cost of the initiator is C^r = \$10^5/m^3. Actually, the parameters for the scheduling problems are all encapsulated in the two parameters A given in eq. (24) and B given in eq. (22) in the objective function (56). They are A = \$190/h^2 and $B = 0.2$ for the given production data. The value of B also denotes the percentage of all transition times over the total cycle time and the feasibility condition of the scheduling problem is $B < 1$.

In the integrated method, the dynamic optimization problems for all potential transitions are first solved offline and they generate a set of controller candidates with the associated transition times and transition costs. The integrated problem is then approximated by a simultaneous scheduling and controller selection problem and it is solved online. In this work, all optimization problems are modeled in GAMS 23.7.3 and solved in a PC with Intel(R) Core(TM) i5-2400 CPU @ 3.10 GHz, 8 GB RAM, and Window 7 64-bit operating system. Though the CPU has 4 cores, only 1 core is used in the computation.

Offline dynamic optimization

We first consider the dynamic optimization of the control problem in each possible transition period. The solution to the dynamic optimization problem is calculated by the discretization method via the orthogonal collocation (Cuthrell & Biegler, 1987). A critical parameter in the method is the number of finite elements and it is set as 75 in this example by the trial over all transitions. The number of finite elements has to be selected large enough for the worst scenario, even though the number is redundant for some other transitions.

To determine the transition time according to the quality bound (37), we need to model both transition and production periods. The finite elements are cast into two sets. The first 45 finite elements (60% of the total) are used to model the process in the transition period while the remaining 30 finite elements (40%) are applied to the production period. The transition time is determined as the end value of the 45$^{\text{th}}$ finite element, which implies $M_t = 45$ in the constraints (37). In each finite element, 3 collocation points are generated, i.e. $N_{cp} = 3$.

Due to the nonlinearity of the process model, the transition time varies considerably for different pairs of the products. Therefore, it is unreasonable to set the same lower bound or upper bound of all transition times. Instead, the same length of the interval is set for the transition time while the starting point and the ending point for each interval are allowed to change. The length of the transition time interval is determined to be 1.5 h in this case.

Table 3
Model and solution statistics for the dynamic optimization which minimizes the transition time in the transition of A→D.

Model	Type NLP	Equations 3,563	Variables 3,380
Solution	Solver CONOPT3	Objective (h) 1.36	CPU (s) 0.7

To generate the controller candidates for selection in the scheduling problem, the interval of the transition time is partitioned equally with the duration of 0.1 h. The interval starts from the minimum transition time with a length equal to 1.5 h and there are 16 partition points. The 16 transition time points for each transition are sufficient for the subsequent integrated problem. The selection of the discrete transition time points can be validated from the objective function value of the integrated problem, which will be presented in the next section after the solution to the integrated problem is calculated. At each partition point except the first one, a dynamic optimization problem is solved with the objective of minimizing the transition cost. The data of the transition time, the transition cost, and the controller parameters returned by dynamic optimization are recorded. The approach is illustrated by the transition from the product A to product D in Fig. 7. The minimum transition time is denoted by $\theta^t_{\min}$ and the corresponding output trajectory is displayed in red. The model and solution statistics for minimizing the transition time are listed in Table 3. Then the transition time is set at $\theta^t_{\min}+0.1$ and the output trajectory (in green) is calculated by minimizing the transition cost. The same procedure is repeated with the transition time equal to $\theta^t_{\min}+0.2$, $\theta^t_{\min}+0.3$, …, $\theta^t_{\min}+1.5$. For clarity, only the output trajectory calculated when the transition time equals the upper bound, $\theta^t_{\min}+1.5$, is displayed in blue.

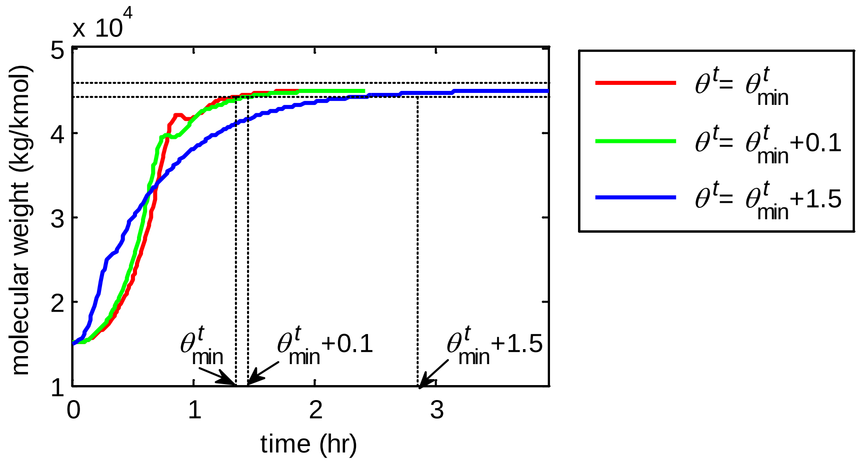


Fig. 7. The output profiles generated by dynamic optimization under different transition times for the transition A→D.

The dynamic optimization is applied to each potential transition A→B, A→C, …, B→A, B→C, …, P→N, P→O. For each transition the dynamic optimization problem is solved at 16 transition points determined in the same way as A→D illustrated above. There are totally 16×15×16=3840 dynamic optimization problems due to the 16 products (the first "16") and the 16 controller candidates (the last "16") for each transition. The total computational time for generating all controller candidates is about 0.9 hour. We note that all these problems are solved offline.

Simultaneous scheduling and controller selection problem

After the dynamic optimization problems are solved, a set of controller candidates for each transition is calculated. The controller parameters as well as the associated transition time and transition cost are stored. The MIDO is approximated by a scheduling problem combined with the controller selection from the candidates provided by dynamic optimization problems. The resulting model is much simpler than the original integrated model and the problem sizes of the two models are listed in Table 4.

The integrated model consists of the scheduling problem and all control problems so there are a great number of equations and variables. Such a large-scale MINLP problem, which is formulated by using 75 finite elements to discretize the dynamic system in each transition, is very difficult to solve in a short time. Even the search for a feasible solution is not trivial so it is formidable to calculate the global optimum. In comparison, the simultaneous scheduling and controller selection problem is much smaller since the equations and variables related to the dynamic model are all excluded. Though more binary variables are included in the model for the controller selection, the problem can be solved efficiently and globally. The comparison of solutions to the two types of model is presented in the following section and the focus here is on the solution strategies for the simultaneous scheduling and controller selection problem.

Table 4
Statistics for the original integrated problem and the simultaneous scheduling and controller selection problem (16 products)

Model	Type	Equations	All Variables	Discrete Variables
Simultaneous scheduling and control	MINLP	57,668	58,548	255
Scheduling and controller selection	MINLP	804	8,452	4,351

The simultaneous scheduling and controller selection problem is a non-convex MINLP problem with a nonlinear fractional objective function and linear constraints. The MINLP solver of DICOPT (Kocis & Grossmann, 1989) cannot guarantee the global optimality of the solution. Though another popular MINLP solver SBB (Bussieck & Drud, 2001) is generally a local one, it can calculate the global optimal solution to the problem because the relaxed NLP problem is quasi-convex as discussed in Section 3.2. To obtain the global optimal solution, the global optimization solvers such as BARON (Tawarmalani & Sahinidis, 2004) can be used as well. An alternative for the global solution is to solve the fractional programming problem via the parametric programming approach presented in the previous section. The bisection method (Fig. 5) or the Dinkelbach's method (Fig. 6) can be applied to solve the equivalent parametric programming problem. For a given parameter, the optimization problem at each iteration step is a convex mixed-integer quadratic programming (MIQCP), which can be solved by CPLEX 12.

The two methods based on the parametric programming are applied first. The iteration procedures in both methods aim to find the root of the optimal-value function and the stopping tolerance is set as $\left|F\left(q_m\right)\right| < 0.1$. The Dinkelbach's method starts from the initial value $q_0 = 0$ and the iteration procedure stops at the 5^{th} step. The optimal value returned is q_5 = \$3,504.0/h and the computational time for all iterations is 7.7 s.

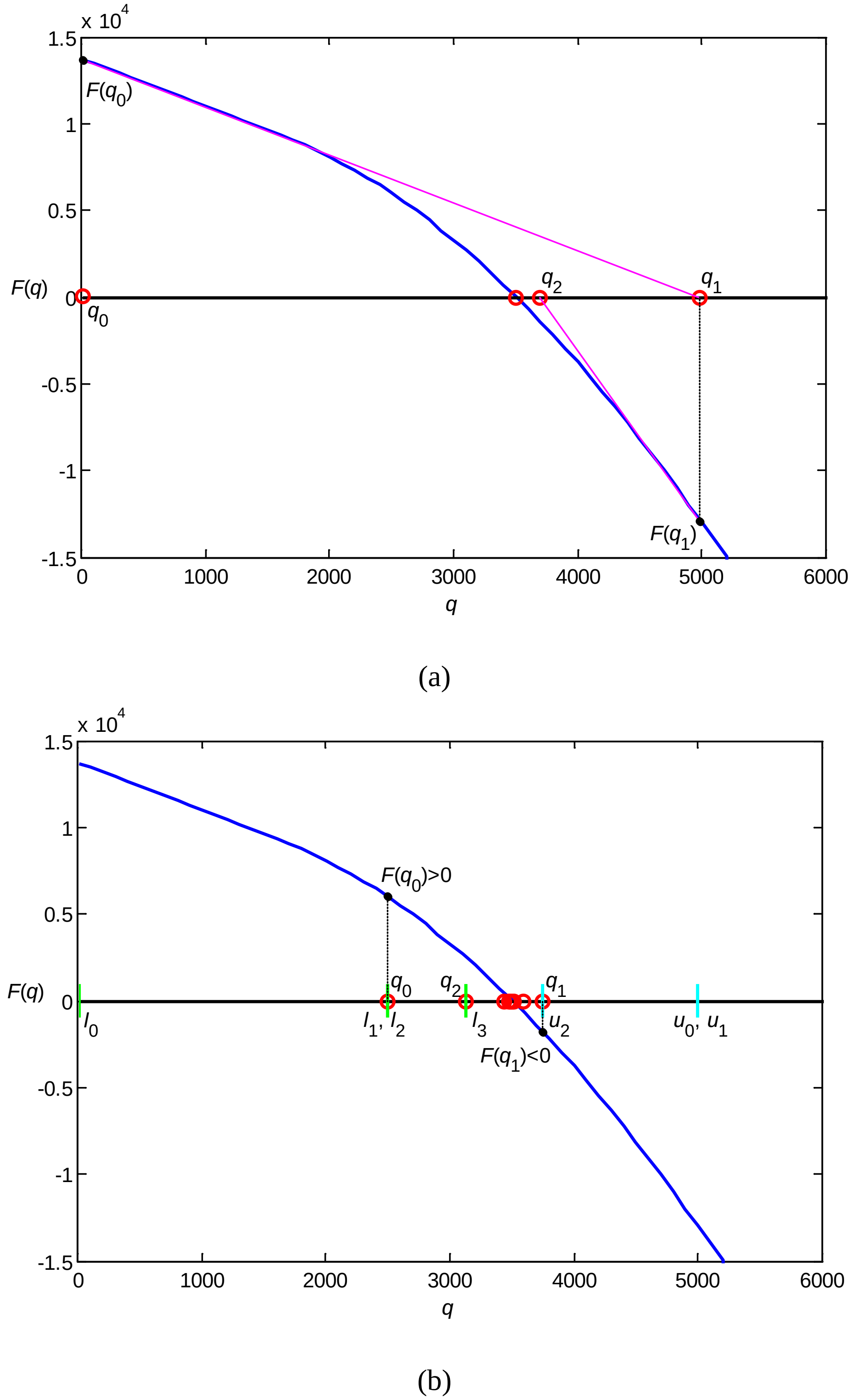
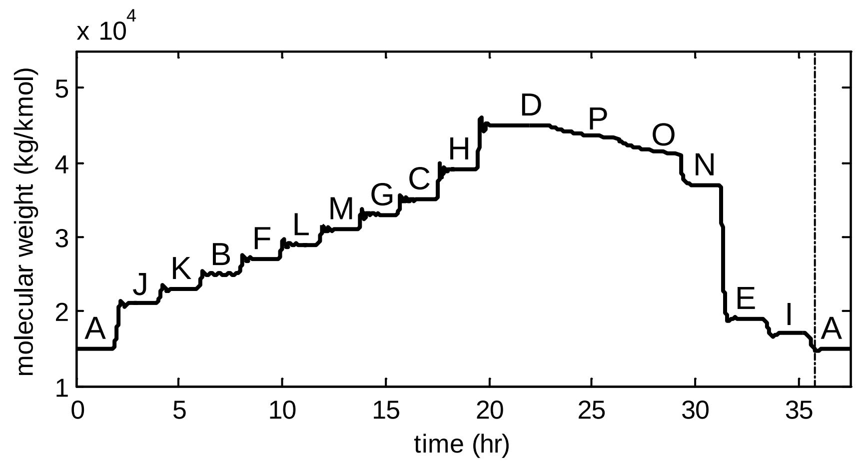
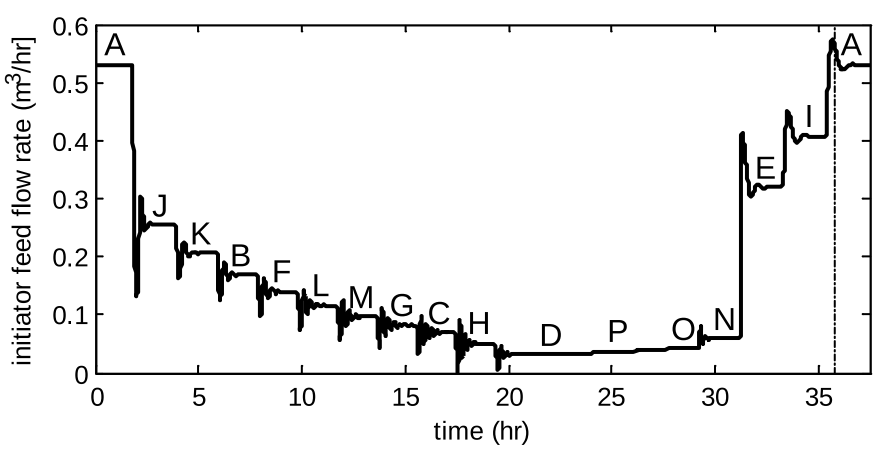


Fig. 8. Iteration procedure of the parametric programming for (a) the Dinkelbach's method and (b) the bisection method.

The iteration procedure is visualized in Fig. 8(a). The optimal-value function $F(q)$ plotted in blue is concave, continuous, and strictly decreasing. There is a unique solution of $F(q) = 0$ and the iteration aims to find the solution. The Dinkelbach's method is based on the Newton's algorithm using the subgradient. The iteration starts from the value of q_0 and the subgradient (in magenta) at the point is calculated. A straight line is created with the slope equal to the subgradient and it intersects the optimal-value function at $F(q_0)$. The function of $F(q)$ is below this line according to the definition of the subgradient. The next value of q_1 is equal to zero of the straight line. We then calculate the subgradient at q_1 to obtain the next value of q_2. The same procedure repeats until the value of $F(q)$ is less than the toleration level.



(a)



(b)

Fig. 9. Optimal scheduling results by the integration method: (a) the output profile and (b) the input profile.

The bisection method sets the initial lower bound $l_0 = 0$, the upper bound $u_0 = 5{,}000$, so the initial parameter value $q_0 = (l_0+u_0)/2 = 2{,}500$. The bisection method uses 18 iteration steps to find the same optimal solution as the Dinkelbach's method and the computational time for all 18 iterations is 16.0 s. The visualization of the iteration procedure is displayed in Fig. 8(b). The iterative value of q is the middle point between the lower bound and the upper bound. The bounds are updated according to the sign of $F(q)$. For example, the function value at the initial point of q_0 is positive so the lower bound (in green) is set as $l_1 = q_0$ while the upper bound (in cyan) is not changed, i.e. $u_1 = u_0$. The new value of q is calculated as $q_1 = (l_1+u_1)/2$ and the function value at this point is negative. So the upper bound is updated as $u_2 = q_1$ while the lower bound keeps the same as $l_2 = l_1$. After one iteration, the range between the lower bound and the upper bound decreases by 50% and the iteration stops when the function value is less than the tolerance level.

The bisection method converges more slowly than the Dinkelbach's method because it only uses the sign information of $F(q)$ while the Dinkelbach's method is able to fully utilize the whole functional value in the iteration. Both methods based on the parametric programming find the global optimal solution. The scheduling results and the dynamic profiles of the input and output are displayed in Fig. 9. In the total cost rate of \$3,504.0/h, the inventory cost rate is \$1,355.8/h and the transition cost rate is \$2,148.1/h. The total cycle time is 35.6 h. When the global solver BARON 9.3.1 is used, it requires 661.8 s to calculate the optimal solution of \$3,505.3/h with the optimality gap set as 0.1%.

It is observed that the input trajectory in Fig. 9(b) is oscillatory at beginning of each transition period. The fluctuations are generated due to the complexity of the polymerization process and the same phenomena are also observed in the literature (Terrazas-Moreno et al., 2008) where the same model is studied. To smooth the input trajectories, a penalty term on the difference between the input values at two adjacent time points can be added to the objective function (49). The penalty term can be the sum of the absolute differences or the squared differences. Due to the penalty on the change of the input, a smoother input trajectory can be obtained (Ozkan et al., 2001). It is beyond the scope of this paper to consider the regularization procedure. However, fluctuations are less noticeable in the output trajectory in Fig. 9(a) and they do not have significant effects on the product quality.

For a thorough comparison, we perform computational experiments by solving a number of instances with the Dinkelbach's method and the bisection method via CPLEX 12, the direct methods via BARON 9.3.1, SBB, and DICOPT respectively. Different values of *A* and *B* in the objective function are set by varying the demand rate D_i. The nominal value of D_i is 0.5 m^3/h and it varies from 0.56 m^3/h to 0.44 m^3/h. Correspondingly, the value of *B*, the percentage of all transition times over the cycle time, changes from 10% to 30% by the step of 5%. The results are summarized in Table 5.

Table 5 (on Page 59)

The Dinkelbach's method is the most computationally efficient. It not only returns the global optimal solution, but also requires much shorter computation times than other methods. The bisection method is slower than the Dinkelbach's method indicated by both iteration numbers and computation times. However, the bisection method is still more efficient than other global methods by BARON 9.3.1 and SBB. The results demonstrate the advantage of the solution via the parametric programming over the direct optimization of the fractional objective function. The global optimizer BARON 9.3.1 guarantees the global optimal solution while it requires significantly longer computational time. The local solver DICOPT requires the smallest computation time among the three direct methods and its computational time is close to the parametric programming methods'. However, DICOPT cannot guarantee the global optimality and only a local solution is found in the case of No. 1. Though, in this example, another local solver of SBB returns the same solutions as the global ones, it requires much more computation time. It is concluded from the comparisons that the Dinkelbach's method has the best performance.

Comparison with the sequential method

Integration of scheduling and control is able to achieve a better overall performance by making a tradeoff between the conflicting factors in both problems. In the objective function (23) the transition times appear in both the numerator and the denominator of the fraction and apparently a good tradeoff is required to minimize the total cost rate. The conventional method, however, solves the two problems sequentially and fails to take the tradeoff into account.

Comparisons of the proposed integration method with the conventional sequential method are visualized in Fig. 10. For each possible transition, the conventional method determines a single controller by minimizing the transition time. The controller determines the process dynamics uniquely and thus the transition time and the transition cost in a transition period are both fixed. Then the scheduling decisions are calculated based on the fixed information in each transition period. In comparison, for each transition, the integration method generates a pool of candidate controllers according to the discretized transition times. Therefore, a set of transition times and a set of corresponding transition costs are calculated. Then the integration method determines the scheduling decisions simultaneously with selection of a controller from the candidate pool for each transition. Obviously, the integration method has more degrees of freedom than the sequential method, which provide opportunities to improve the performance of the production process.

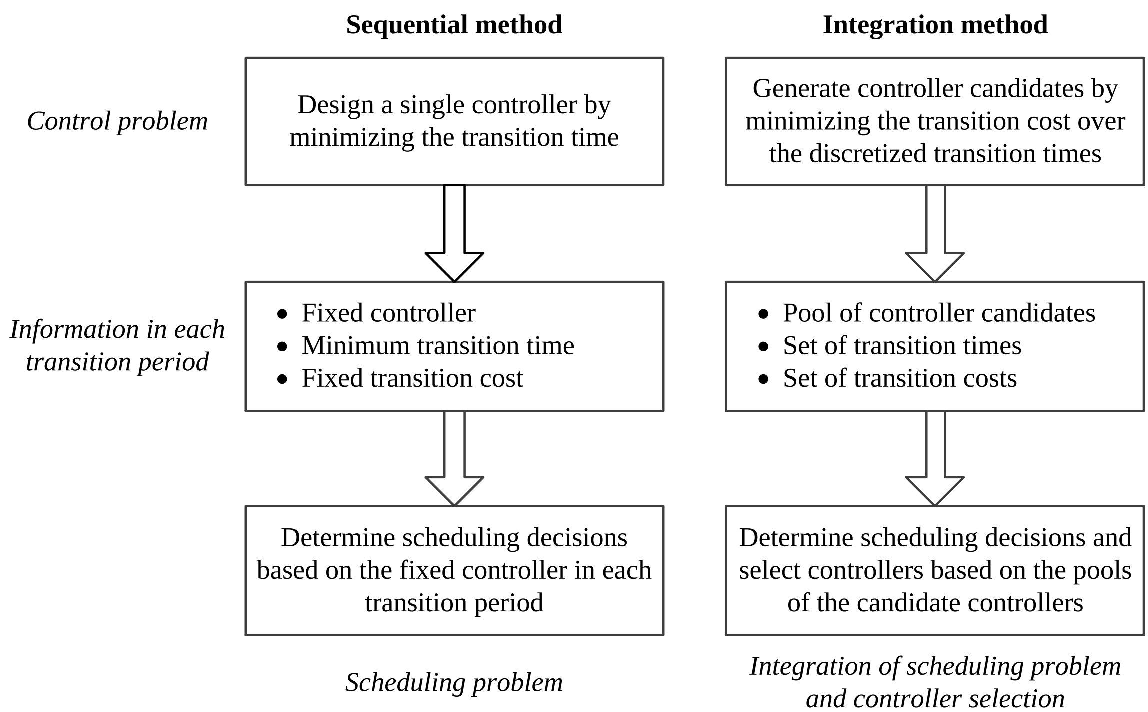
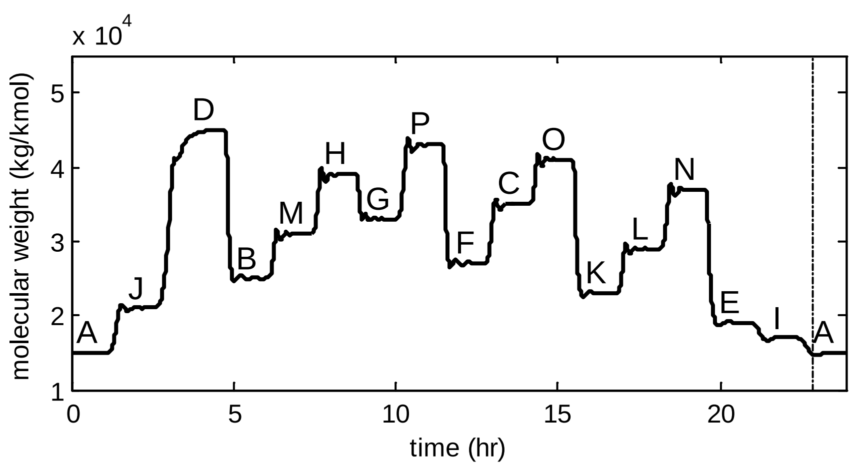
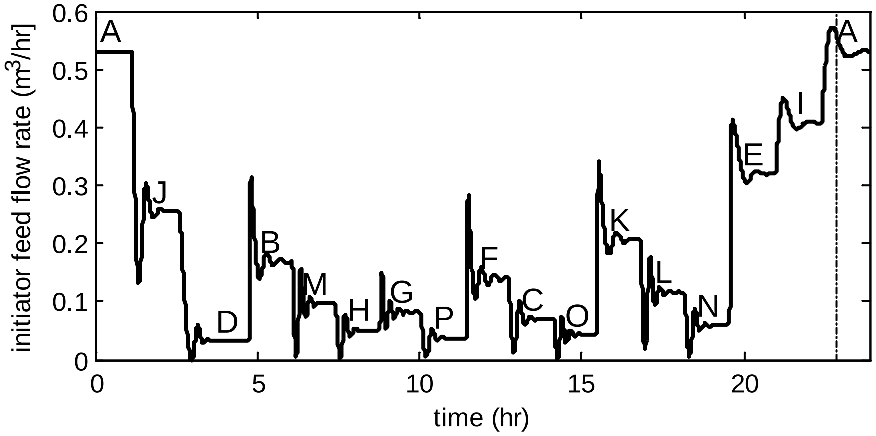


Fig. 10. Comparison of the proposed integration method with the conventional the sequential method

The sequential method can be regarded as a special case of the scheduling and controller selection problem where only one controller candidate corresponding to the minimum transition time is considered for every possible transition. There is no degree of freedom for the scheduling problem to adjust the process dynamics. The scheduling results returned by the sequential method are displayed in Fig. 11. The production sequence returned by the sequential method is distinct from the integration method (Fig. 9) and the total cycle time (22.7 h) is much smaller than that (35.6 h) in the integration method due to the minimum transition time. The objective function value for the sequential method is $4,192.8/h, which is about 19.7% larger than the optimal one, $3,504.0/h, returned by the integrated strategy. The inventory cost rate returned by the sequential method is $863.8/h and it is less than $1,355.8/h returned by the integration method. However, the transition cost rate returned by the sequential method is $3,328.9/h which is much larger than $2,148.1/h returned by the integration method. The inventory cost rate is proportional to the total transition time while the transition cost rate increases as the transition time decreases. Therefore, the minimum transition time does not imply the minimum total cost rate. The integration method can well balance the two cost rates and result in a smaller total cost rate than the sequential method.



(a)



(b)

Fig. 11. Optimal scheduling results by the sequential method: (a) the output profile and (b) the input profile.

Besides the two end-point values of N_C = 1 or N_C = 16, the simultaneous scheduling and controller selection problem can also be solved with other selections of N_C. The changes in the objective function value as the number of controller candidates for each transition increases are plotted in Fig. 12. When only one controller candidate exists, it is the controller that causes the minimum transition time. When n controller candidates are used, they are created by the procedure discussed in the previous subsection with the transition time ranging from the minimum one to the minimum one plus 0.1•(n−1) h. The largest number of controller candidates is set as 16.

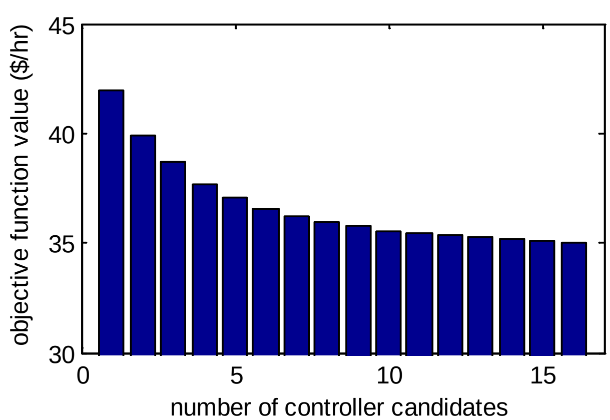


Fig. 12. Change in the objective function value as the number of controller candidates increases.

The objective function value decreases with the number of controller candidates for each transition. However, the improvement becomes marginal when the number of controller candidates is large. For example, there is a less than 0.1% improvement when the number of candidate controllers increases from 15 to 16. Therefore the 16 candidate controllers for each transition are sufficient for this case study. It is seen that even the selection of two controller candidates is able to reduce the total cost rate by 5% compared with the case of the single controller candidate. This, again, demonstrates the advantage of integrating the scheduling problem and the control problem so that a better performance can be achieved.

Comparison with the simultaneous method

A common solution to the MIDO for the integrated problem is reformulating it into an MINLP (Flores-Tlacuahuac & Grossmann, 2006, 2011) by the full discretization of the differential equations (Biegler, 2007; Cuthrell & Biegler, 1987). The discretized dynamic models in all periods are solved simultaneously with the scheduling problem. The method is straightforward and an MINLP solver is applied to solve the integrated problem (39) directly without further analysis and decomposition of the model. All complexities are taken care of by the MINLP solver. The main disadvantage of this method is that the resulting problem could be potentially very large-scale, which might be quite computationally expensive. For the purpose of comparison, the simultaneous method is applied to the same production model and the results are compared with the proposed method of scheduling and controller selection.

Table 6
Comparison of the proposed method with the simultaneous method

Model	4 products, A-D		8 products, A-H	
Method	simultaneous	proposed	simultaneous	proposed
Equations	14,324	60	28,708	212
All Variables	13,632	340	27,676	1,604
Dis. Variables	15	207	63	959
Objective ($/h)	2,332.9	2,334.1	2,730.3	2,725.0
CPU (s)	43.3	0.8	24,330	1.2

The dynamic optimization for the simultaneous method adopts the same settings as the one applied to generate the controller candidates in the proposed method, e.g. 75 finite elements for discretizing the continuous-time trajectories of the states, input and output. For the scheduling problem of all 16 products, a large-scale MINLP problem is derived and the model statistics can be seen in Table 4. There are 57,668 equations and 58,548 variables and the problem cannot be solved by SBB with a reasonable time (12 hours). To make the simultaneous method applicable, small-scale models are derived with fewer products. The models with 4 products and 8 products are subtracted from the model of 16 products. For each model, the proposed method and the simultaneous method are applied and the results are listed in parallel in Table 6. For the simultaneous method, the solver of SBB is applied to solve the reformulated MINLP with gap set to 1% while the Dinkelbach's method using the MIQCP solver CPLEX 12 with 0% gap is used for the simultaneous scheduling and controller selection problem.

In the case of 4 products, the proposed method solves the problem in less than one second and the optimal objective function value is \$2,334.1/h. The simultaneous method requires 43.3 s in calculating the optimal value of \$2,332.9/h which is a little bit smaller than that returned by the proposed method. The two methods return the same production sequence but different transition times. The difference stems from the discretization resolution of the transition time in the proposed method where the transition time is discretized with the step of 0.1 h. The values of the transition time returned by the two methods are listed in Table 7. It is seen that the proposed method selects the transition times which are closest (with the discretization resolution of 0.1 h) to those by the simultaneous method. The discretization affects the optimal solution, however, the effect is negligible (0.05%).

Table 7
Values of the transition time (h) returned by the proposed method and the simultaneous method. The model is the 4-product manufacture and transition sequence is A→B→D→C for both methods.

Method	A→B	B→D	D→C	C→A
Proposed	0.44	1.89	0.12	0.41
Simultaneous	0.44	1.89	0.14	0.41

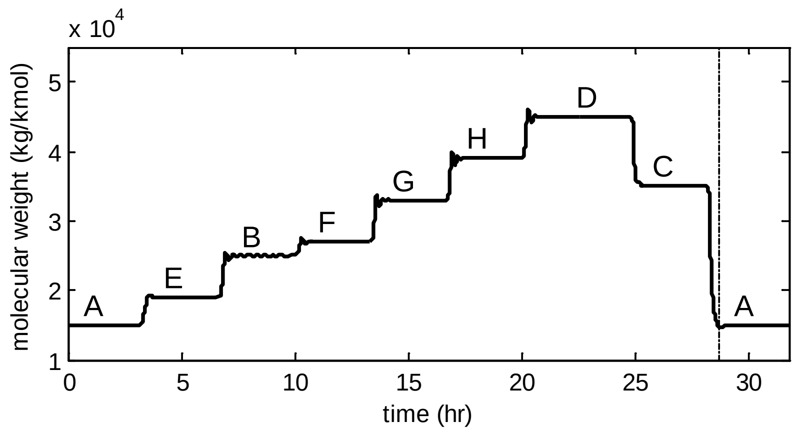
In the case of 8 products, the advantage of the proposed method in the computational efficiency becomes more evident. It requires only 1.2 s in calculating the optimal objective function value of \$2,725.0/h while the simultaneous approach requires more than 6.7 hours to obtain the optimal value of \$2,730.3/h. The simultaneous method returns a little bit worse solution (due to the 1% gap) than the proposed method even though the computation time is 4 orders of magnitude larger. The computational time of the simultaneous method is even much larger than the total time of the proposed method to which the time of offline dynamic optimization is added. There are $8 \times 7 \times 16 = 896$ dynamic optimization problems which are solved offline to generate the controller candidates for the scheduling problem and the total computation time is about 10 minutes. The inefficiency of the simultaneous method arises from the strong interactions between the dynamic optimization in a slot and those in other slots. The interaction becomes much stronger when the number of products increases so the computational time for the 8-product case is much longer than the 4-product case.

Based on the comparison, it is concluded that the proposed method is able to get the equivalent solution to the simultaneous method and the error resulted from the discretization of the transition time is marginal. However, the proposed method is much more computationally efficient than the simultaneous method, especially when more products are scheduled. The efficiency is due to the decomposition of the dynamic optimization problems from the scheduling problem. The model complexity is significantly decreased after the decomposition. All constraints become linear and the relaxed problem is a quasi-convex fractional programming. Therefore, tailored global optimization algorithms can be easily applied.

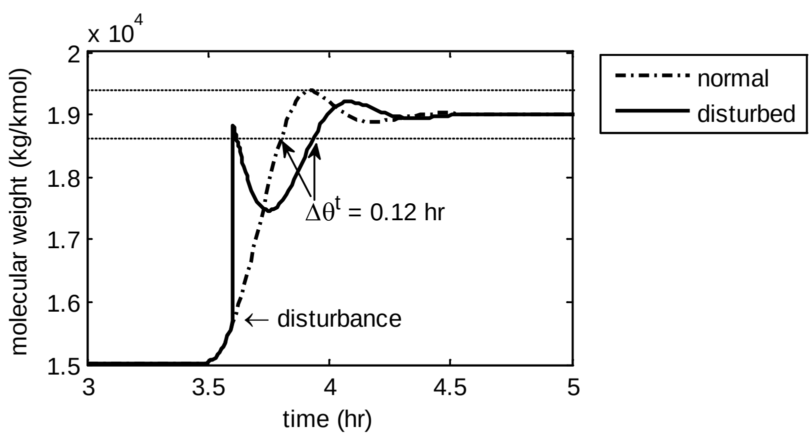
Online implementation of the integrated method

Uncertainties and disturbances are inevitable in the production process and it is necessary to re-solve the integrated problem online to deal with them. The efficiency of the proposed method enables the online solution of the simultaneous scheduling and controller selection problem. Although the proposed method can be implemented in the case of 16 products, the case of 8 products is used to illustrate the online rescheduling for the purpose of a clear visualization of the results. The process initially follows scheduling results without any disturbances, which are shown in Fig. 13(a).

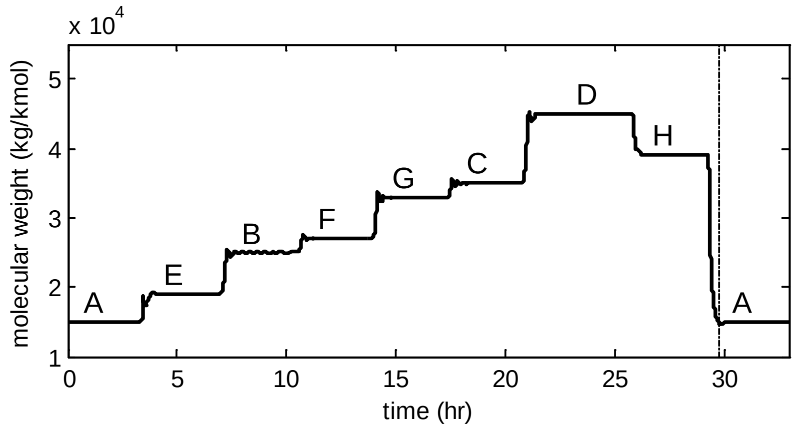
A disturbance is supposed to take place in the first transition period from the product A to the product E as displayed in Fig. 13(b). The disturbance deviates the output trajectory from the normal one. Due to the PI controller, the disturbance can be automatically attenuated and the output can finally enter the production range for the product E without solving the integrated problem. This is an advantage of the proposed framework where the integrated problem is solved to update the controller parameters rather than the control variable. If the value of the control variable is calculated directly from the integrated problem, the integrated problem needs to be solved immediately after the disturbance appears. Otherwise, the control loop will work in the open loop. By contrast, the proposed method updates the controller parameters so even if the integrated problem is not solved the output can also be controlled to the setpoint value. The approach of updating the controller parameters is favored considering the issues of production safety.



(a)



(b)



(c)

Fig. 13. Online rescheduling under the disturbance. (a) The normal scheduling results without disturbances; (b) The occurrence of the disturbance on the output; (c) The online re-scheduling results under the disturbances.

Though the output can be controlled to track the setpoint by the controller itself, the process still needs to be rescheduled under the disturbance. The disturbance extends the transition time and consequently the cycle time is prolonged. If the process is not rescheduled, the production rate will be less than the demand rate and an infeasible solution is generated. The re-scheduling is performed at the beginning of the production period of the product E and the re-scheduled results are shown in Fig. 13(c). The variables in the periods before the rescheduling are fixed by the existing data, e.g. the assignment of A and E to the first two slots, the production time of A and the transition time from A to E after the disturbance. Other variables are updated by re-solving the simultaneous scheduling and controller selection problem. The re-scheduling does not only change the production time and the transition time but also changes the production sequence. The cycle time increases from 28.6 h in the offline solution to 29.6 h under the disturbance and the total cost rate increases from $2,725.0/h to $2,809.2/h.

In the integration method, only controllers in transition periods are determined simultaneously with scheduling decisions. The controller in a production period can be simply adopted as the one in the previous transition period. The controller can be also designed based on the operating condition of the production period. Since the process is in the steady state during the production period, controller design for the production period is much simpler than that for the transition period. Because the process model can be linearized around the steady state, standard design methods can be applied (Ang et al., 2005; Skogestad, 2003). In addition, controller design for the production period is not dependent on the production sequence so the controllers can be designed separately from scheduling decisions. Due to these reasons, controller design for the production period is not taken into account in the integration method.

We assume the production period is controlled by a PI controller no matter if it follows the one in the previous transition period or it is designed separately from scheduling decisions. The only requirement is that it is able to attenuate a disturbance and drive the process back to the steady state after the disturbance. Then the updated information is used to recalculate the integrated problem online to determine the new schedule along with controller parameters in the transition periods. The disturbance occurring during the production period is dealt with in the same way as the disturbance in the transition period.

Besides the uncertainties in the control system, the production uncertainties, e.g. the variation in the material cost or the change in the order demand, are frequently encountered. Facing the dwindling profit margin caused by the stronger global competition, the flexible production which allows the system to react to the changes in the environment becomes increasingly important.

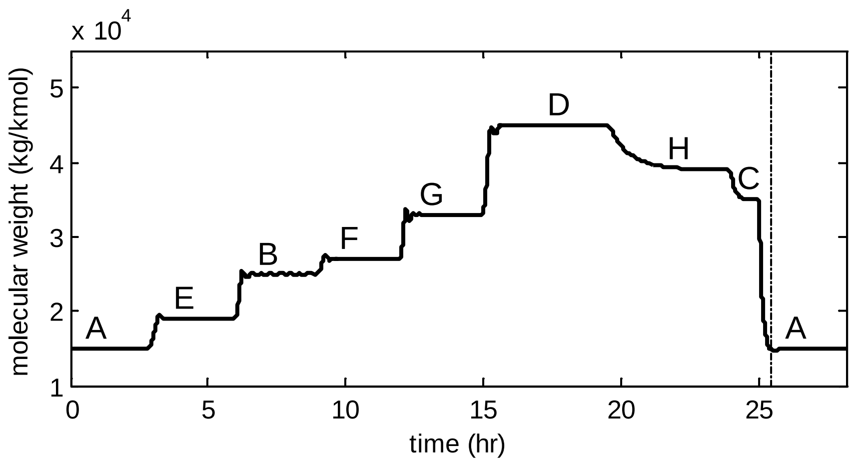


Fig. 14. Online rescheduling under the change in order demand.

For example, a rush order cancellation is assumed to be received and the demand rate of the product C decreases by 70% after the production of A. Similar to the case of the disturbance in the control system, the integrated problem is re-solved online. The rescheduling results are shown in Fig. 14. The production sequence changes and the production period of the product C shrinks due to the decrease of its demand. The total cost rate changes to $3,007.6/h and the cycle time becomes 25.3 h.

In the proposed method, the dynamic optimization problems are decomposed from the integrated problem and there is no need to re-calculate them again under the production uncertainties. The decomposition method also benefits the solution under the change in the production recipe, e.g. adding or removing a product. Considering that the process model which describes the reaction mechanism is seldom changed compared with the production recipe and the production parameters, the decomposition procedure of calculating and storing the information from the dynamic model in advance is more favorable than the simultaneous method. In the simultaneous approach, the entire mixed-integer dynamic optimization problem has to be solved again when the scheduling parameters change.

5. Conclusion

A novel method for integrating the scheduling problem and the control problem is proposed. The process is assumed to be controlled by the PI controller and the controller parameters are calculated simultaneously with the scheduling decisions in the integrated problem. By exploring the model structure, the dynamic optimization for the control problems is decomposed from the scheduling problem. The dynamic optimization for each transition is solved independently and a set of controller candidates are created via discretization of the transition time. The MIDO for the integrated problem is approximated by a simultaneous scheduling and controller selection problem. The derived problem is a mixed-integer nonlinear fractional program with linear constraints, which can be globally optimized in an efficient way using the Dinkelbach's algorithm.

The computational results demonstrate that the proposed integration method is much more efficient than the simultaneous method which has to solve both control and scheduling problems at the same time. In a problem with 8 products, the proposed method calculates the global solution to the scheduling and control selection problem in seconds while the simultaneous method require several hours to solve the integrated problem directly. Though the scheduling and controller selection problem is an approximation of the integrated problem by discretizing the transition time, the solution is equivalent to that returned by directly solving the integrated problem. The efficiency becomes more evident as the number of products increases. In a problem with 16 products, the simultaneous method fails to return a feasible solution while the proposed method returns the solution in about ten seconds. The efficiency enables the proposed method to be implemented online to deal with various uncertainties and disturbances, e.g. the deviation of the output or the change in the order demand.

Nomenclature

Indices

i, j	products
k	time slots
l, l'	controller candidates
m	finite elements in collocation method
n	iteration steps in parametric programming
p, r	collocation points

Variables

c_T	total transition cost ($)
$d(t)$	integrated control error
$e(t)$	control error
$F(q)$	optimal-value function in parametric programming
h	length of finite element (h)
h_k	length of finite element in slot k (h)
K_I	controller parameter of integral term (Note 1)
K_P	controller parameter of proportional term (Note 1)
l_n	lower bound of q_n
q	coefficient in parametric programming
q_n	value of q in iteration n
u_n	upper bound of q_n
t_c	cycle time (h)
t_T	total transition time (h)

$u(t)$ material flow rate/control variable/process input (m^3/h)

$v(t)$ integrated material flow rate (m^3)

v_{mp} discretized $v(t)$ at p-th collocation point in m-th finite element

w_i produced amount of product i (m^3)

$x(t)$ state variable in process model

$x^c(t)$ state variable in closed-loop control system

x_0^c initial value of $x^c(t)$

$x_0^{c,k}$ initial value of $x^c(t)$ in slot k

x_{mp}^c discretized $x^c(t)$ at p-th collocation point in m-th finite element

$x_{m,0}^c$ value at starting collocation point in m-th finite element

$\dot{x}_{mp}^c$ discretized $\dot{x}^c(t)$ at p-th collocation point in m-th finite element

$y(t)$ process output

y_{sp} setpoint of process output

y_{sp}^k setpoint of process output in slot k

y_{mp} discretized $y(t)$ at p-th collocation point in m-th finite element

$z(t)$ augmented variable in in closed-loop control system

z_{mp} discretized $z(t)$ at p-th collocation point in m-th finite element

Greek Letter

β_{ijk} 0-1 variable to denote if product j is after product i in slot k

η_k collection of variables in control problem in slot k

δ_k^p material cost in production period of slot k (\$)

δ_k^t material cost in transition period of slot k (\$)

δ_{ij}^t material cost in transition from product i to product j (\$)

δ_{ijl}^t material cost in transition from product i to product j by controller l (\$) (Note 1)

γ_{ijl} 0-1 variable to denote if controller l is selected for transition from product i to product j

φ objective function of total cost rate (\$/h)

φ_1 inventory cost rate (\$/h)

φ_2 material cost rate (\$/h)

Θ_i production time of product i (h)

θ_k^p production time in slot k (h)

θ_k^t transition time in slot k (h)

θ_{ij}^t transition time from product i to product j (h)

θ_{ijl}^t transition time from product i to product j by controller l (h) (Note 1)

$\theta_{ij,\min}^t$ minimum transition time from product i to product j (h)

ξ_{ik} 0-1 variable to denote if product i is assigned to slot k

Parameters

A coefficient in objective function

B coefficient in objective function

C_i^s inventory cost rate of product i [\$/(m^3•h)]

C^r raw material cost (\$/m^3)

C_{rp} collocation matrix

D_i demand rate of product i (m^3/h)

D_θ discretization resolution in transition time (h)

E_i steady-state value of $x^c(t)$ for manufacturing product i

F_i steady-state value of $u(t)$ for manufacturing product i

G_i production rate of production i (m^3/h)

H_i steady-state value of $y(t)$ for manufacturing product i

M_t index of finite element for transition time

N number of products/slots

N_{cp} number of collocation points

N_C number of controller candidates

Note 1: They are variables in the dynamic optimization problem as well as the integrated MIDO problem while they are parameters in the scheduling and controller selection problem.

Reference

Allgor, R. J., & Barton, P. I. (1999). Mixed-integer dynamic optimization I: problem formulation. Computers & Chemical Engineering, 23, 567-584.

Ang, K. H., Chong, G., & Li, Y. (2005). PID control system analysis, design, and technology. IEEE Transactions on Control Systems Technology, 13, 559-576.

Astrom, K. J., Panagopoulos, H., & Hagglund, T. (1998). Design of PI controllers based on non-convex optimization. Automatica, 34, 585-601.

Bansal, V., Sakizlis, V., Ross, R., Perkins, J. D., & Pistikopoulos, E. N. (2003). New algorithms for mixed-integer dynamic optimization. Computers & Chemical Engineering, 27, 647-668.

Biegler, L. T. (2007). An overview of simultaneous strategies for dynamic optimization. Chemical Engineering and Processing, 46, 1043-1053.

Bradley, J. R., & Arntzen, B. C. (1999). The simultaneous planning of production, capacity, and inventory in seasonal demand environments. Operations Research, 47, 795-806.

Bussieck, M. R., & Drud, A. S. (2001). SBB: A new solver for mixed integer nonlinear programming. In.

Cambini, A., Crouzeix, J. P., & Martein, L. (2002). On the pseudoconvexity of a quadratic fractional function. Optimization, 51, 677-687.

Capon-Garcia, E., Moreno-Benito, M., & Espuna, A. (2011). Improved Short-Term Batch Scheduling Flexibility Using Variable Recipes. Industrial & Engineering Chemistry Research, 50, 4983-4992.

Cuthrell, J. E., & Biegler, L. T. (1987). On the optimization of differential-algebraic process systems. AIChE Journal, 33, 1257-1270.

Dinkelbach, W. (1967). On Nonlinear Fractional Programming. Management Science, 13.

Duran, M. A., & Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical Programming, 36, 307-339.

Embirucu, M., Lima, E. L., & Pinto, J. C. (1996). Survey of advanced control of polymerization reactors. Polymer Engineering and Science, 36, 433-447.

Flores-Tlacuahuac, A., & Grossmann, I. E. (2006). Simultaneous cyclic scheduling and control of a multiproduct CSTR. Industrial & Engineering Chemistry Research, 45, 6698-6712.

Flores-Tlacuahuac, A., & Grossmann, I. E. (2011). Simultaneous cyclic scheduling and control of tubular reactors: Parallel production lines. Industrial & Engineering Chemistry Research, 50, 8086-8096.

Grossmann, I. (2005). Enterprise-wide optimization: A new frontier in process systems engineering. AIChE Journal, 51, 1846-1857.

Harjunkoski, I., Nystrom, R., & Horch, A. (2009). Integration of scheduling and control-Theory or practice? Computers & Chemical Engineering, 33, 1909-1918.

Huang, Y. J., & Wang, Y. J. (2000). Robust PID tuning strategy for uncertain plants based on the Kharitonov theorem. ISA Transactions, 39, 419-431.

Kim, T. H., Maruta, I., & Sugie, T. (2008). Robust PID controller tuning based on the constrained particle swarm optimization. Automatica, 44, 1104-1110.

Kocis, G. R., & Grossmann, I. E. (1989). Computational experience with DICOPT solving MINLP problems in process systems engineering. Computers & Chemical Engineering, 13, 307-315.
Levner, E., Kats, V., de Pablo, D. A. L., & Cheng, T. C. E. (2010). Complexity of cyclic scheduling problems: A state-of-the-art survey. Computers & Industrial Engineering, 59, 352-361.
Luus, R. (1990). Application of dynamic programming to high-dimensional non-linear optimal control problems. International Journal of Control, 52, 239-250.
Mahadevan, R., Doyle, F. J., & Allcock, A. C. (2002). Control-relevant scheduling of polymer grade transitions. AIChE Journal, 48, 1754-1764.
McAuley, K. B., & Macgregor, J. F. (1992). Optimal grade transitions in a gas phase polyethylene reactor. AIChE Journal, 38, 1564-1576.
Mendez, C. A., Cerda, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers & Chemical Engineering, 30, 913-946.
Misener, R., & Floudas, C. A. (2012). GloMIQO: Global mixed-integer quadratic optimizer. Journal of Global Optimization.
Mishra, B. V., Mayer, E., Raisch, J., & Kienle, A. (2005). Short-term scheduling of batch processes. A comparative study of different approaches. Industrial & Engineering Chemistry Research, 44, 4022-4034.
Mitra, K., Gudi, R. D., Patwardhan, S. C., & Sardar, G. (2010). Resiliency issues in integration of scheduling and control. Industrial & Engineering Chemistry Research, 49, 222-235.
Munoz, E., Capon-Garcia, E., Moreno-Benito, M., Espuna, A., & Puigjaner, L. (2011). Scheduling and control decision-making under an integrated information environment. Computers & Chemical Engineering, 35, 774-786.
Nie, Y., Biegler, L. T., & Wassick, J. M. (2012). Integrated scheduling and dynamic optimization of batch processes using state equipment networks. AIChE Journal, in press.
Nystrom, R. H., Franke, R., Harjunkoski, I., & Kroll, A. (2005). Production campaign planning including grade transition sequencing and dynamic optimization. Computers & Chemical Engineering, 29, 2163-2179.
Nystrom, R. H., Harjunkoski, I., & Kroll, A. (2006). Production optimization for continuously operated processes with optimal operation and scheduling of multiple units. Computers & Chemical Engineering, 30, 392-406.
Ozkan, G., Ozen, S., Erdogan, S., Hapoglu, H., & Alpbaz, M. (2001). Nonlinear control of polymerization reactor. Computers & Chemical Engineering, 25, 757-763.
Pinto, J. M., & Grossmann, I. E. (1994). Optimal cyclic scheduling of multistage continuous multiproduct plants. Computers & Chemical Engineering, 18, 797-816.
Prata, A., Oldenburg, J., Kroll, A., & Marquardt, W. (2008). Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. Computers & Chemical Engineering, 32, 463-476.
Quesada, I., & Grossmann, I. E. (1992). An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. Computers & Chemical Engineering, 16, 937-947.
Rawlings, J. B., Maravelias, C. T., & Subramanian, K. (2012). Integration of control theory and scheduling methods for supply chain management. In Foundations of Computer-Aided Process Operations (FOCAPO) 2012. Savannah, Georia USA.

Sahinidis, N. V., & Grossmann, I. E. (1991). MINLP model for cyclic multiproduct scheduling on continuous parallel lines. Computers & Chemical Engineering, 15, 85-103.
Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. Journal of Process Control, 13, 291-309.
Soylemez, M. T., Munro, N., & Baki, H. (2003). Fast calculation of stabilizing PID controllers. Automatica, 39, 121-126.
Srinivasan, B., Palanki, S., & Bonvin, D. (2003). Dynamic optimization of batch processes - I. Characterization of the nominal solution. Computers & Chemical Engineering, 27, 1-26.
Subramanian, K., Maravelias, C. T., & Rawlings, J. B. (2012). State-space model for chemical production scheduling. Computers & Chemical Engineering.
Tan, N., Kaya, I., Yeroglu, C., & Atherton, D. P. (2006). Computation of stabilizing PI and PID controllers using the stability boundary locus. Energy Conversion and Management, 47, 3045-3058.
Tawarmalani, M., & Sahinidis, N. V. (2004). Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Mathematical Programming, 99, 563-591.
Tawarmalani, M., & Sahinidis, N. V. (2005). A polyhedral branch-and-cut approach to global optimization. Mathematical Programming, 103, 225-249.
Terrazas-Moreno, S., Flores-Tlacuahuac, A., & Grossmann, I. E. (2008). Simultaneous design, scheduling, and optimal control of a methyl-methacrylate continuous polymerization reactor. AIChE Journal, 54, 3160-3170.
Toscano, R. (2005). A simple robust PI/PID controller design via numerical optimization approach. Journal of Process Control, 15, 81-88.
Vassiliadis, V. S., Sargent, R. W. H., & Pantelides, C. C. (1994). Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. Industrial & Engineering Chemistry Research, 33, 2111-2122.
Voudouris, V. T., & Grossmann, I. E. (1993). Optimal synthesis of multiproduct batch plants with cyclic scheduling and inventory considerations. Industrial & Engineering Chemistry Research, 32, 1962-1980.
Westerlund, T., Skrifvars, H., Harjunkoski, I., & Porn, R. (1998). An extended cutting plane method for a class of non-convex MINLP problems. Computers & Chemical Engineering, 22, 357-365.
Wolsey, L. A. (1997). MIP modelling of changeovers in production planning and scheduling problems. European Journal of Operational Research, 99, 154-165.
You, F. Q., Castro, P. M., & Grossmann, I. E. (2009). Dinkelbach's algorithm as an efficient method to solve a class of MINLP models for large-scale cyclic scheduling problems. Computers & Chemical Engineering, 33, 1879-1889.
Zhuge, J., & Ierapetritou, M. (2012). Simultaneous scheduling and control with closed loop implementation on parallel units. In Foundations of Computer-Aided Process Operations (FOCAPO) 2012. Savannah, Georia USA.

Table 5

Solutions to the simultaneous scheduling and controller selection problem

No	D_i	A	B	Global optimization method										Local method	
				Dinkelbach's			Bisection			BARON 9.3.1		SBB		DICOPT	
				Iter.	Obj.	CPU (s)	Iter.	Obj.	CPU (s)	Obj.	CPU (s)	Obj.	CPU (s)	Obj.	CPU (s)
1	0.56	424.7	0.10	4	3309.1	5.2	17	3309.1	18.9	3309.6	642.4	3309.9	95.6	3339.6	25.7
2	0.53	268.3	0.15	5	3395.5	9.4	17	3395.5	24.8	3395.6	391.1	3395.6	415.0	3395.5	26.6
3	0.50	190.0	0.20	5	3504.0	7.7	18	3504.0	16.0	3505.3	661.8	3505.8	286.6	3504.0	22.0
4	0.47	143.0	0.25	5	3645.8	10.7	16	3645.9	20.4	3646.9	675.3	3646.4	148.8	3645.8	18.3
5	0.44	111.6	0.30	5	3817.7	11.3	19	3817.7	20.1	3818.3	284.5	3817.7	254.0	3818.3	21.7

For the Dinkelbach's method and the bisection method, the optimality gap of the MIQCP solved by CLEPX 12 in each iteration step is set as 0.0%.

For the solvers of BARON 9.3.1, SBB, and DICOPT which solve the problem directly, the optimality gap of the MINLP is set as 0.1%.

SBB returns the global solution since the relaxed NLP problem is quasi-convex where the local solution is also the global one.