Computers and Chemical Engineering 58 (2013) 315–333

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Integration of production scheduling and dynamic optimization for multi-product CSTRs: Generalized Benders decomposition coupled with global mixed-integer fractional programming

Yunfei Chu, Fengqi You*

Department of Chemical & Biological Engineering, Northwestern University, Evanston, IL 60208, USA

ARTICLE INFO

Article history:
Received 15 November 2012
Received in revised form 29 May 2013
Accepted 2 August 2013
Available online 13 August 2013

Keywords:
Production scheduling
Dynamic optimization
Generalized Benders decomposition
Fractional programming
Polymerization process

ABSTRACT

Integration of production scheduling and dynamic optimization can improve the overall performance of multi-product CSTRs. However, the integration leads to a mixed-integer dynamic optimization problem, which could be challenging to solve. We propose two efficient methods based on the generalized Bender decomposition framework that take advantage of the special structures of the integrated problem. The first method is applied to a time-slot formulation. The decomposed primal problem is a set of separable dynamic optimization problems and the master problem is a mixed-integer nonlinear fractional program. The master problem is then solved to global optimality by a fractional programming algorithm, ensuring valid Benders cuts. The second decomposition method is applied to a production sequence formulation. Similar to the first method, the second method uses a fractional programming algorithm to solve the master problem. Compared with the simultaneous method, the proposed decomposition methods can reduce the computational time by over two orders of magnitudes for a polymer production process in a CSTR.

1. Introduction

Scheduling and dynamic optimization are two important decision levels in process engineering (Engell & Harjunkoski, 2012; Grossmann, 2005; Wassick et al., 2012). Traditionally, the two problems are solved sequentially. The dynamic optimization problem is solved for every possible transition to determine the operational conditions, which provide the recipe data for the scheduling problem. These recipe data are treated as fixed parameters when the production schedule is optimized. However, it has been recently demonstrated that a collaborative optimization approach which solves the integrated scheduling and dynamic optimization problem simultaneously can significantly improve the overall performance of the entire process system because the operational conditions can be optimized along with the production sequence and assignments (Bassett et al., 1996; Munoz, Capon-Garcia, Moreno-Benito, Espuna, & Puigjaner, 2011; Chu & You, 2013b).

The integrated scheduling and dynamic optimization problem is generally formulated as a mixed-integer dynamic optimization (MIDO) problem (Bansal, Sakizlis, Ross, Perkins, & Pistikopoulos, 2003; Barton, Allgor, Feehery, & Galan, 1998; Mohideen, Perkins, & Pistikopoulos, 1996) or a mixed-logic dynamic optimization (MLDO) problem (Oldenburg, Marquardt, Heinz, & Leineweber, 2003; Prata, Oldenburg, Kroll, & Marquardt, 2008). An MLDO problem can be transformed into an MIDO problem by big-M constraints or a convex-hull reformulation (Turkay & Grossmann, 1996). A common solution strategy for the integrated MIDO problem is the simultaneous method (Flores-Tlacuahuac & Grossmann, 2006; Mishra, Mayer, Raisch, & Kienle, 2005; Terrazas-Moreno, Flores-Tlacuahuac, & Grossmann, 2007). This method discretizes the differential equations by a collocation method (Biegler, 2007) or a Runge–Kutta method (Ascher, Ruuth, & Spiteri, 1997). After the discretization procedure, the MIDO problem is reformulated as a mixed-integer nonlinear programming (MINLP) problem. A general-purpose MINLP solver can be used to solve the MINLP for the integrated problem. Though the simultaneous method is straightforward, the reformulated MINLP problem is usually large-scale and can be very challenging to solve (Harjunkoski, Nystrom, & Horch, 2009; Klatt & Marquardt, 2009).

The objective of this work is to develop fast optimization algorithms to solve the integrated scheduling and dynamic optimization problem for a multi-product CSTR. To reduce the computational complexity, we present systematic decomposition methods based on the framework of the generalized Benders decomposition (GBD) (Floudas, 1995; Geoffrion, 1972). The proposed methods are outlined in Fig. 1.

* Corresponding author. Tel.: +1 847 467 2943; fax: +1 847 491 3728.
E-mail address: you@northwestern.edu (F. You).

0098-1354/$ – see front matter
http://dx.doi.org/10.1016/j.compchemeng.2013.08.003

Nomenclature

Abbreviation

GBD generalized Benders decomposition
MIDO mixed-integer dynamic optimization
MILP mixed-integer linear programming
MINLFP mixed-integer nonlinear fractional programming
MINLP mixed-integer nonlinear programming
MIQCP mixed-integer quadratic programming
MLDO mixed logic dynamic optimization
MMA methyl methacrylate
NLP nonlinear programming

Optimization problem

Fractional P general fractional programming problem
Integration PS integrated problem formulated by production sequence model
Integration TS integrated problem formulated by time-slot model
Master I master problem of decomposition method I
Master II master problem of decomposition method II
Minimize TT dynamic optimization problem for minimizing transition time
Parametric I equivalent parametric programming problem of Master I
Parametric II equivalent parametric programming problem of Master II
Parametric P general parametric programming problem
Primal I primal problem of decomposition method I
Primal II primal problem of decomposition method II

Index

i product
j product
k time slot
m iteration step in generalized Benders decomposition
n iteration step in generalized Benders decomposition
r iteration step in fractional programming algorithm

Binary variable

β_{ij} equal to one if product i precedes product j
ξ_{ik} equal to one if time slot k is assigned to product i
B_{ijk} equal to one if product i is manufactured in time slot k before product j

Greek Letter

σ_i variable for subtour elimination
δ_{ij} transition cost from product i to product j
$\delta_{ij}^{\max}$ upper bound of δ_{ij}
Δ_k transition cost in time slot k
ε threshold value in generalized Benders decomposition
ε_D threshold value in fractional programming algorithm
φ total cost rate
$\varphi_{\text{Inventory}}$ inventory cost rate
$\varphi_{\text{Transition}}$ transition cost rate
$\varphi_{\text{Production}}$ production cost rate
Ψ_k copy of transition time in time-slot k
λ_k Lagrange multiplier in decomposition method I
λ_{ij} Lagrange multiplier in decomposition method II
μ objective function value of master problems
π_k Lagrange multiplier in decomposition method I
ρ_k Lagrange multiplier in decomposition method I
τ_{ij} copy of transition time from product i to product j
θ_{ij} transition time from product i to product j
$\theta_{ij}^{\min}$ lower bound of θ_{ij}
$\theta_{ij}^{\max}$ upper bound of θ_{ij}
Θ_k transition time in time slot k
Θ_k^p production time in time slot k
ζ_{ik} relaxation of assignment variable ξ_{ik}

Variable

BT_{ij} product of β_{ij} and θ_{ij}
BD_{ij} product of β_{ij} and δ_{ij}
H_k length of finite element in time slot k
T_c cycle time
U_{ij} input variable of dynamic system in transition from product i to product j
U_k input variable of dynamic system in time slot k
X_{ij} state variable of dynamic system in transition from product i to product j
X_k state variable of dynamic system in time slot k
X_k^0 initial condition of dynamic system in time slot k
Y_k output variable of dynamic system in time slot k
Y_{ij} output variable of dynamic system in transition from product i to product j
Y_k^{sp} setpoint value of dynamic system in time slot k

Parameter

c^r unit price of initiator
d_i demand rate of product i
g_i production rate of product i
q parameter in fractional programming algorithm
x_i^0 steady-state value of dynamic system for product i
y_i^{sp} setpoint value of dynamic system for product i

We focus on the integrated scheduling and dynamic optimization problem for continuous processes in a CSTR, where multiple products are manufactured in a cyclic manner (Pinto & Grossmann, 1994). Though it cannot cover all integrated problems, e.g. batch production (Chu & You, 2013a; Mishra et al., 2005; Nie, Biegler, & Wassick, 2012), the formulation is widely used to build an integrated problem for continuous production (Flores-Tlacuahuac & Grossmann, 2006; Nystrom, Franke, Harjunkoski, & Kroll, 2005; Prata et al., 2008; Terrazas-Moreno et al., 2007; Zhuge & Ierapetritou, 2012).

It should be noted that GBD is a framework which can produce variant decomposition methods. In this work, we propose two decomposition methods based on two formulations of the integrated problem. The purpose for proposing the multiple methods instead of a single one is to determine a better solution strategy in practice. The performance of a decomposition method generally depends on specific data for the scheduling problem and particular models for the dynamic systems. Due to the variety and complexity of an integrated problem, it is difficult to guarantee that a decomposition method is always superior to the other. Developing different decomposition methods provides an opportunity for one to make a comparison according to a specific problem.

The key idea of GBD is to identify the complicating variables. When they are temporarily fixed, the restricted problem is much easier to solve than the original problem. To identify the complicating variables, exploring the problem structure is essential. Since the integrated scheduling and dynamic optimization problem can be formulated as an MINLP, a special structure of the integrated model is the coupling of the binary variables and nonlinear equations.

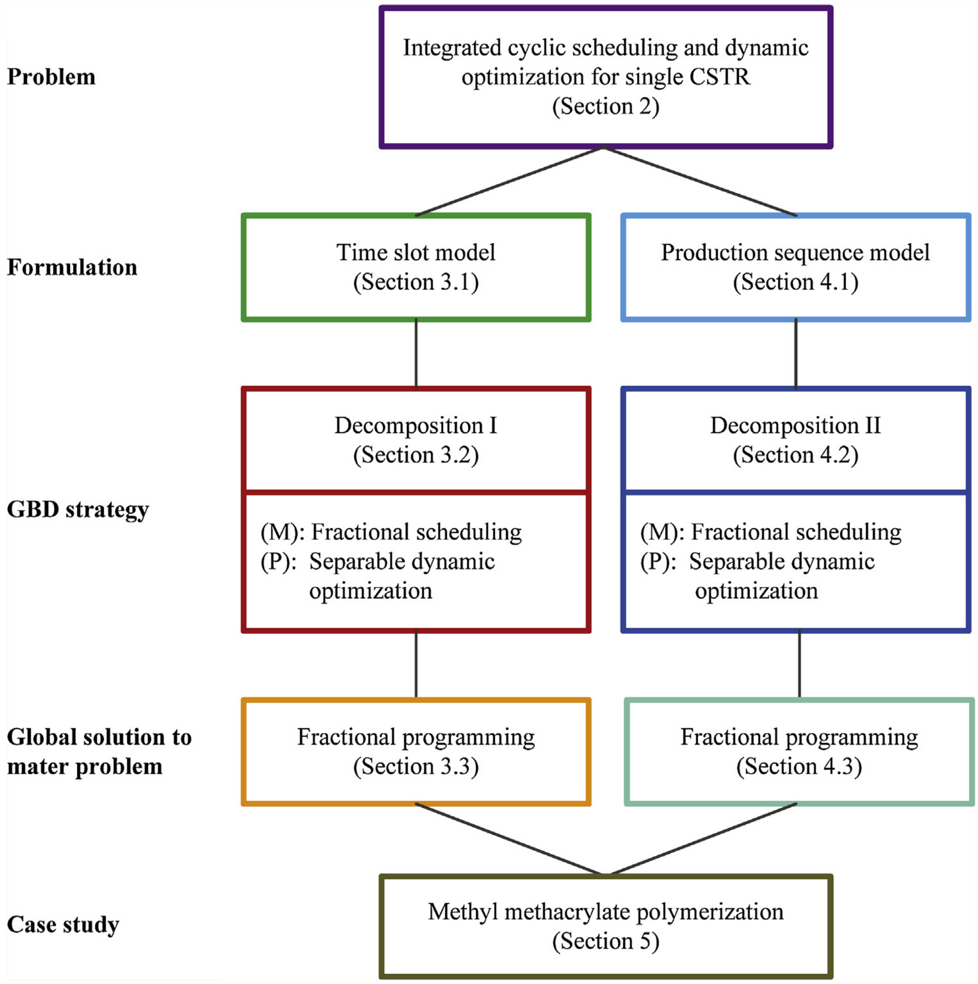

Fig. 1. The outline of the proposed methods as well as the roadmap of the paper. In the GBD methods, (M) denotes the mater problem and (P) denotes the primal problem.

Classifying the binary variables as the complicating ones is a common approach in a GBD method (Bansal et al., 2003).

However, the integrated problem has more special structures we can exploit to design more effective GBD methods. Another special structure is the interface between the scheduling model and the dynamic models. The scheduling model provides the initial condition and the setpoint for each dynamic model while the dynamic models return the transition times and the transition costs to the scheduling model (Chu & You, 2012; Flores-Tlacuahuac & Grossmann, 2006; Nystrom et al., 2005; Prata et al., 2008; Zhuge & Ierapetritou, 2012). By breaking the linking equations, we develop two new decomposition methods. In both methods, the primal problem is a set of separable dynamic optimization problems, and the master problem becomes a mixed-integer nonlinear fractional programming (MINLFP) problem. The master problems still have special structures which allow its global optimization using efficient fractional programming algorithms. Global optimality guarantees valid Benders cuts.

The novel contributions of this work are summarized as

- A formulation of the integrated scheduling and dynamic optimization problem for a multi-product CSTR based on the production sequence model
- Two decomposition methods (referred to as method I and method II) which are able to exploit special structures of the integrated MIDO problem
- Efficient global solution approaches for the master problems, providing valid Benders cuts

The two decomposition methods are demonstrated in a case study of a methyl methacrylate polymerization process. Compared with the popular simultaneous method which solves the integrated problem directly by discretizing the differential equations describing the dynamic models, the proposed decomposition methods can reduce the computational time by more than two orders of magnitudes.

As integrated scheduling and dynamic optimization problems are many and various, we cannot demonstrate the proposed methods for all cases. The restrictions of the proposed methods should be justified. First, the methods are developed for the cyclic production in a CSTR with the specifications stated in Section 2. Second, the convergence of the GBD decomposition methods could be affected by the non-convexity of dynamic models. Not only for the decomposition method, the non-convexity is also a difficulty for a direct solution approach, e.g. the simultaneous method. Because an integrated problem includes a number of dynamic models rather than a single one, and each dynamic model can be complicated, the non-convexity is difficult to deal with in a general case. However, we propose two decomposition methods. Comparing the results by different methods could provide a practical way to check the convergence and choose the most suitable method according to a particular problem.

2. Problem statement

Considering there is a great variety of scheduling problems, a single model cannot cover all problems. In this work, we focus on the cyclic scheduling model in a single multi-product CSTR, which is the most commonly used scheduling model to build the integrated problem (Chatzidoukas, Pistikopoulos, & Kiparissides, 2009; Chu & You, 2012; Flores-Tlacuahuac & Grossmann, 2006; Nystrom et al., 2005; Prata et al., 2008; Zhuge & Ierapetritou, 2012). Specifically, the integrated problem we investigate is stated as follows:

Production model:
Cyclic production in a single multi-product CSTR
Given:
(*Scheduling parameters*)
Production rate and demand rate for each product
Unit cost for inventory and utility
(*Dynamic models*)
Differential equations characterizing process dynamics
Operational conditions for each product
Determine:
Production sequence
Cycle time
Transition time and transition cost
Dynamic trajectories of input, output, and state variables
Objective:
Minimize *total cost = inventory cost + transition cost*

3. Time-slot formulation and solution method

In this section, we first formulate the integrated scheduling and dynamic optimization problem for a multi-product CSTR based on the time-slot model in Section 3.1. Then the GBD method is presented in Section 3.2. The method is referred to as decomposition method I in the paper. The primal problem consists of a set of separable dynamic optimization problems over the time slots. The master problem is an MINLFP which can be solved to global optimality in Section 3.3.

3.1. Formulation based on time-slot assignment model

As illustrated in Fig. 2, multiple products A, B, C, and D are produced one by one in a multi-product CSTR. Accordingly, the time interval of the CSTR is partitioned into a set of consecutive time slots. Each product is produced in a time slot only once and each time slot is used to produce only one product. Thus, the number of the time slots is identical to the number of the products. The main scheduling decision in the time-slot model is to assign the time slots to the products. The assignment is determined by a set of binary variables $\xi_{ik} \in \{0, 1\}$. If $\xi_{ik} = 1$, then time-slot k is assigned to product i, as illustrated in Fig. 2.

Using the assignment variables, the integrated problem is formulated as (Chu & You, 2012)

$$(\textit{Integration TS}) \quad \min\ a_1 \sum_{k=1}^{n_p} \Theta_k + a_2 \frac{\sum_{k=1}^{n_p} \Delta_k}{\sum_{k=1}^{n_p} \Theta_k} \tag{1}$$

s.t.

Scheduling model

$$\xi_{ik} \in \{0, 1\},\ \forall i,\ k \tag{2}$$

$$\zeta_{ik} = \xi_{ik},\ \forall i,\ k \tag{3}$$

$$\zeta_{ik} \in [0, 1],\ \forall i,\ k \tag{4}$$

$$\sum_{k=1}^{n_p} \zeta_{ik} = 1,\ \forall i \tag{5}$$

$$\sum_{i=1}^{n_p} \zeta_{ik} = 1,\ \forall k \tag{6}$$

Dynamic models

$$\frac{dX_k(t)}{dt} = F(X_k(t), U_k(t)),\ \forall k \tag{7}$$

$$Y_k(t) = G(X_k(t), U_k(t)),\ \forall k \tag{8}$$

$$H(X_k(t), U_k(t), Y_k(t)) \le 0,\ \forall k \tag{9}$$

$$X_k(0) = X_k^0,\ \forall k \tag{10}$$

$$Y_k(t) = Y_k^{sp},\ \forall t \ge \Psi_k,\ \forall k \tag{11}$$

$$\Delta_k = \phi_k(X_k(\Psi_k), U_k(\Psi_k), Y_k(\Psi_k)),\ \forall k \tag{12}$$

$$\Psi_k = \Theta_k,\ \forall k \tag{13}$$

Linking equations

$$X_k^0 = \sum_{i=1}^{n_p} \zeta_{ik} x_i^0,\ \forall k \tag{14}$$

$$Y_k^{sp} = \sum_{i=1}^{n_p} \zeta_{ik++1} y_i^{sp},\ \forall k \tag{15}$$

The objective function (1) consists of the averaged inventory cost and the averaged transition cost in a cycle, where Θ_k and Δ_k represent the transition time and the transition cost in time slot k. When there is no overproduction, the total production time is a fixed ratio of the cycle time, and the averaged production cost is a constant, dropped from the objective function (Chu & You, 2012). The coefficients in the objective function are

$$a_1 = \sum_{i=1}^{n_p} \frac{c_i^I (g_i - d_i) d_i / 2g_i}{\left(1 - \sum_{i=1}^{n_p} d_i/g_i\right)}$$

$$a_2 = 1 - \sum_{i=1}^{n_p} \frac{d_i}{g_i}$$

where the parameter c_i^I is the unit inventory cost of product i, g_i is the production rate, and d_i is the demand rate.

To facilitate the subsequent decomposition method, we introduce a copy of the binary variables, denoted by ζ_{ik}. They are continuous variables ranging from zero to one. However, they can only have the binary value according to the equality constraint (3). The constraints (5) and (6) ensure one product is manufactured only once in a time slot and a time slot is used to produce only one product. The dynamic models in time slots are formulated as constraints (7)–(13). The dynamic models are indexed by k and there are n_p dynamic models we need to consider. In each dynamic model, the state variables are represented by $X_k(t)$, the inputs by $U_k(t)$, and the

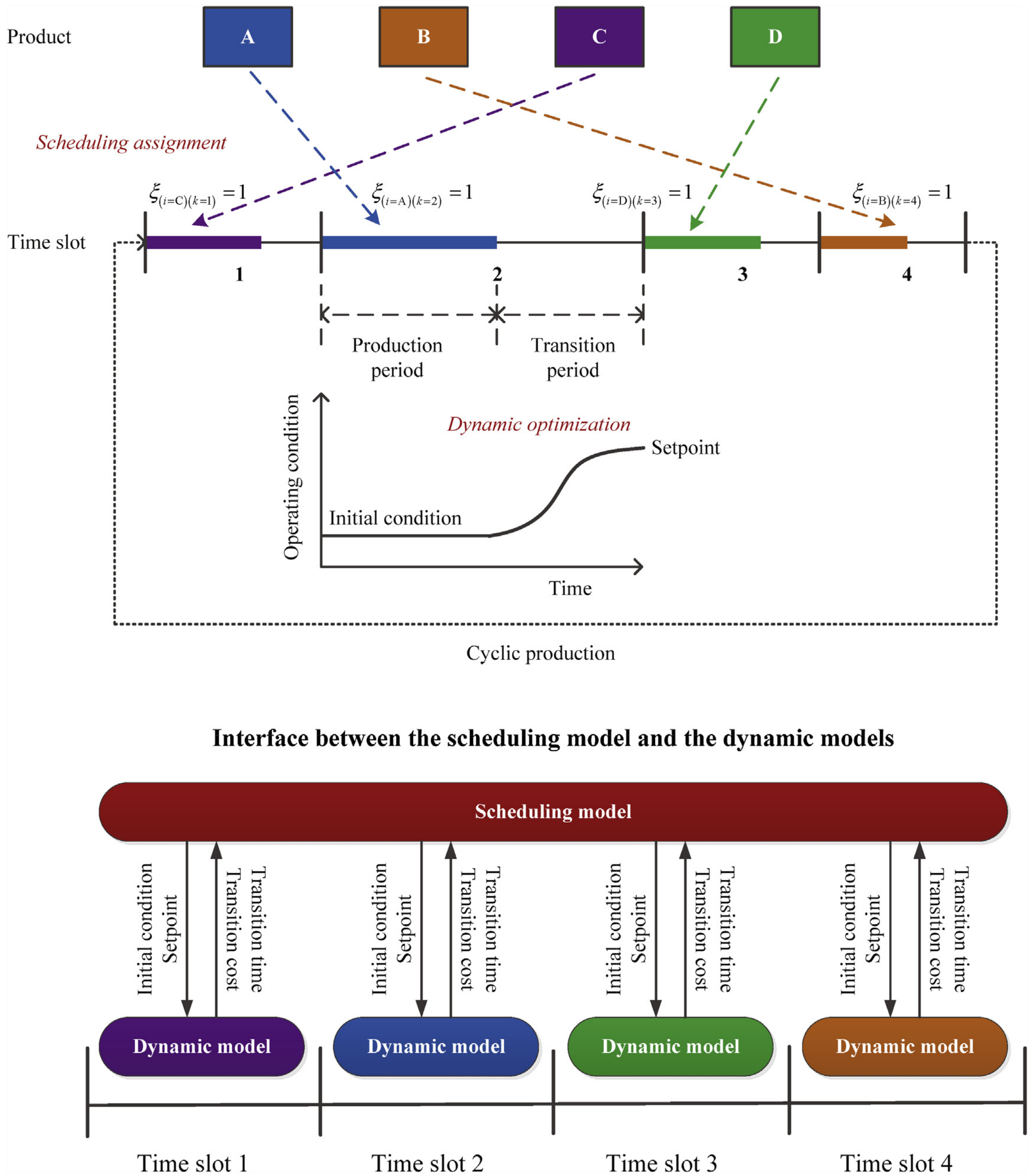

Fig. 2. Illustration of the integrated problem for a multi-product CSTR based on the time-slot model.
The scheduling assignment is determined by the binary variable ξ_{ik}, which is one when product i is assigned to time-slot k. The transition period of each time slot is governed by a dynamic model. On one hand, the initial condition and the setpoint is determined by the scheduling model according to the value of the assignment variables. On the other hand, manipulating the input variables of the dynamic model can determine the transition time and the transition cost in a transition period.

outputs by $Y_k(t)$. To have a compact expression, we use the vector notation. Each state, input, or output vector can include multiple elements. The vector notation is applied to the equations as well. The differential equation (7) represents the systems equation and the equation (8) determines the outputs. Inequality (9) imposes path constraints on the states, inputs and outputs. The initial condition is specified in Eq. (10) and the setpoint value is given in Eq. (11). The transition time Θ_k is defined as the length of the time interval from the starting point of the transition to the ending point after which the output stays at the setpoint value. For the purpose of the decomposition methods, we introduce a copy of the transition time, denoted by Ψ_k. The consensus equality (13) forces Ψ_k to have an identical value with Θ_k. Eq. (12) gives the definition of the transition cost Δ_k. For different processes, we have different expressions of the transition cost.

In the time-slot model, the dynamic models are coupled with the binary scheduling variables through Eqs. (14) and (15). These equations define the interface between the scheduling model and the dynamic models, which is visualized in Fig. 2. The assignment variable is used to determine the initial condition and the setpoint value of each dynamic system. The steady-state value x_i^0 and the setpoint value y_i^{sp} for a product are given parameters. However, the initial condition X_k^0 and the setpoint value Y_k^{sp} in a time slot are variables depending on the assignment decisions. Because the production is carried out cyclically, the cyclic addition $k{++}1$ is used in Eq. (15), which is a shortcut notation for

$$\xi_{ik{++}1} = \begin{cases} \xi_{ik+1}, & 1 \le k \le n_p - 1 \\ \xi_{i1}, & k = n_p \end{cases}, \quad \forall i$$

Due to the linking equations and the consensus equality between ζ_{ik} and ξ_{ik}, the assignment variable ξ_{ik} has a strong effect on the dynamic system. When ξ_{ik} varies, the initial condition and the setpoint are changed and in turn all variables regarding the dynamic models are changed.

3.2. Method I: decomposition for time-slot formulation

Direct solution to the complicated MIDO problem (*Integration TS*) can be computationally expensive. A decomposition method is helpful to reduce the computational time. In this section, we present a GBD method to solve the integrated problem.

GBD has been widely used in process engineering to solve difficult optimization problems (Li, Chen, & Barton, 2012; Sahinidis & Grossmann, 1991; Sakizlis, Perkins, & Pistikopoulos, 2004). Generally, it consists of several main steps (Floudas, 1995; Geoffrion, 1972)

(1) Identify the complicating variables. The presence of these variables makes the optimization problem difficult to solve. However, when they are temporarily fixed, the optimization problem with the remaining variables turns out to be much easier to solve.
(2) Project the optimization problem, including the objective function and the constraints, onto the space of the complicating variables. The projected objective function is determined implicitly by solving an inner optimization problem for the remaining variables.
(3) Decompose the optimization problem into a primal problem and a master problem according to the projection. The primal problem is solved when the complicating variables are fixed. Then the master problem is solved to update the complicating variables based on the dual information returned by the primal problem.
(4) Solve the original problem by iterating between the primal problem and the master problem. The primal problem updates the upper bound of the optimal value while the master problem provides the lower bound. The iteration stops when the gap between the lower bound and the upper bound is less than a specified threshold.

The GBD method has many variants depending on specific decomposition procedures, selection of the 'complicated' variables, formulations of the master and primal problems, the solution algorithms to each subproblem, and so on. In a common GBD approach for solving an MIDO problem, the binary variables are labeled as the complicating ones (Bansal et al., 2003). The decomposition method results in a mixed-integer linear programming (MILP) master problem with Benders constraints and a coupled dynamic optimization primal problem. The primal problem can be, however, difficult to solve because an integrated problem often includes a great number of dynamic models which should be optimized simultaneously in the primal problem.

Considering the primal problem is solved repeatedly in each iteration step for GBD, it is desirable to further simplify the complicated dynamic optimization problem. The integrated problem actually has more special structure, enabling more effective decomposition strategies. In this section, we study a new decomposition method where the complicating variables include not only the binary assignment variables ξ_{ik} but also the continuous variables of the transition times Θ_k. After we fix the value of these variables, we can optimize a dynamic model for each transition period separately.

According to the classification of the complicating variables, the projected objective function is

$$v_I\left(\left\{\bar{\xi}_{ik}\right\}, \left\{\bar{\Theta}_k\right\}\right) = \min\left(a_1\sum_{k=1}^{n_p}\bar{\Theta}_k + a_2\frac{\sum_{k=1}^{n_p}\Delta_k}{\sum_{k=1}^{n_p}\bar{\Theta}_k}\right) \tag{16}$$

$$= a_1\sum_{k=1}^{n_p}\bar{\Theta}_k + a_2\frac{\min\sum_{k=1}^{n_p}\Delta_k}{\sum_{k=1}^{n_p}\bar{\Theta}_k} \tag{17}$$

$$= a_1\sum_{k=1}^{n_p}\bar{\Theta}_k + a_2\frac{\sum_{k=1}^{n_p}\min\ \Delta_k}{\sum_{k=1}^{n_p}\bar{\Theta}_k} \tag{18}$$

When both $\left\{\xi_{ik}\right\}$ and $\left\{\Theta_k\right\}$ are fixed at $\left\{\bar{\xi}_{ik}\right\}$ and $\left\{\bar{\Theta}_k\right\}$ respectively, the first term in the objective function is fixed. The denominator of the second term is fixed as well. Therefore, we only need to minimize the numerator of the second term and Eq. (16) is equivalent to Eq. (17). Further, when the assignment variables $\left\{\xi_{ik}\right\}$ are fixed at $\left\{\bar{\xi}_{ik}\right\}$, the initial condition and the setpoint of a dynamic system are known. If the transition time Θ_k is also known as $\bar{\Theta}_k$, the transition cost Δ_k of a dynamic system can be optimized independently. Minimizing the sum of the transition costs in Eq. (17) is equivalent to the sum of the minimum transition costs in Eq. (18).

In the projected problem, the dynamic models are optimized independently. Thus, the primal problem becomes a series of separated dynamic optimization problems as

$$(Primal\,I)\quad \Delta_k^* = \min\Delta_k, \forall k$$

s.t.
Dynamic model (7)–(12)

$$\Psi_k = \bar{\Theta}_k \tag{19}$$

$$X_k^0 = \sum_{i=1}^{n_p}\bar{\xi}_{ik}x_i^0 \tag{20}$$

$$Y_k^{sp} = \sum_{i=1}^{n_p}\bar{\xi}_{ik{++}1}y_i^{sp} \tag{21}$$

The consensus equality (13) in the integrated problem is replaced by Eq. (19) with the fixed transition time. Similarly, the linking equations (14) and (15) are substituted by constraints (20) and (21), where the assignment variables are fixed. These equations are used to provide the dual information to update the master problem.

By relaxing Eqs. (19)–(21), the projected objective function is expressed by the dual information as

$$v_I\left(\{\bar{\xi}_{ik}\},\{\bar{\Theta}_k\}\right)$$

$$= a_1\sum_{k=1}^{n_p}\bar{\Theta}_k + a_2\frac{\sum_{k=1}^{n_p}\max_{\lambda_k,\rho_k,\pi_k}\min\left[\Delta_k+\lambda_k\left(\bar{\Theta}_k-\Psi_k\right)+\rho_k\left(\sum_{i=1}^{n_p}\bar{\xi}_{ik}x_i^0-X_k^0\right)+\pi_k\left(\sum_{i=1}^{n_p}\bar{\xi}_{ik++1}y_i^{sp}-Y_k^{sp}\right)\right]}{\sum_{k=1}^{n_p}\bar{\Theta}_k}$$

$$= a_1\sum_{k=1}^{n_p}\bar{\Theta}_k + a_2\frac{\sum_{k=1}^{n_p}\max_{\lambda_k,\rho_k,\pi_k}\left[\min\left(\Delta_k-\lambda_k\Psi_k-\rho_kX_k^0-\pi_kY_k^{sp}\right)+\lambda_k\bar{\Theta}_k+\rho_k\sum_{i=1}^{n_p}\bar{\xi}_{ik}x_i^0+\pi_k\sum_{i=1}^{n_p}\bar{\xi}_{ik++1}y_i^{sp}\right]}{\sum_{k=1}^{n_p}\bar{\Theta}_k} \tag{22}$$

where λ_k, ρ_k, and π_k are Lagrange multipliers of Eqs. (19)–(21) in the primal problem, respectively.

According to the dual formulation, the master problem is expressed as

$$(Master\ I)\quad \mu^* = \min a_1\sum_{k=1}^{n_p}\Theta_k + a_2\frac{\sum_{k=1}^{n_p}\Delta_k}{\sum_{k=1}^{n_p}\Theta_k}$$

s.t.

Assignment constrains (2)–(6)

$$\Delta_k \geq \bar{l}_k^{(m)} + \bar{\lambda}_k^{(m)}\Theta_k + \bar{\rho}_k^{(m)}\sum_{i=1}^{n_p}\xi_{ik}x_i^0 + \bar{\pi}_k^{(m)}\sum_{i=1}^{n_p}\xi_{ik++1}y_i^{sp},\qquad 1\leq m\leq n-1,\ \forall k \tag{23}$$

$$\Theta_k \geq \sum_{i=1}^{n_p}\sum_{j=1}^{n_p}\xi_{ik}\xi_{jk++1}\theta_{ij}^{\min},\ \forall k \tag{24}$$

where m and n denote the iteration numbers. Eq. (23) represents the Benders cut constraints. We can calculate its value according to the variant-2 algorithm of GBD (Floudas, 1995). The value of $\bar{l}_k^{(m)}$ is

$$\bar{l}_k^{(m)} = \Delta_k^* - \lambda_k^*\tau_k^* - \rho_k^*X_k^{0*} - \pi_k^*Y_k^{sp*},\ \forall k \tag{25}$$

where Δ_k^*, τ_k^*, X_k^{0*}, and Y_k^{sp*} belong to the optimal solution to the primal problem (*Primal I*), λ_k^*, ρ_k^*, and π_k^* are optimal dual values regarding Eqs. (19)–(21) in the primal problem. The dual variables are the parameters in the master problem, expressed as $\bar{\lambda}_k^{(m)}$, $\bar{\rho}_k^{(m)}$, and $\bar{\pi}_k^{(m)}$ respectively.

Inequality (24) represents the projected feasible range of the master problem. To ensure that the dynamic optimization in the primal problem has a feasible solution, the transition time should not exceed the minimum value. The minimum transition time is present due to the bounded process input. In practice, we cannot produce an input beyond the instrumental limitation so we cannot reduce the transition time to any small value we want. Therefore, the transition time should be bounded from below. However, there is no upper bound of the transition time and we can make the transition period as long as desired.

The minimum transition time is characterized by the dynamic model for each pair of products and it can be determined in advance by solving the dynamic optimization problem

$$(Minimize\ TT)\quad \theta_{ij}^{\min} = \min\theta_{ij},\ \forall i,j$$

s.t.

Dynamic models (44)–(50)

Again the special structure of the integrated scheduling and dynamic optimization problem is exploited to determine the feasible range of the master problem. Though it is possible to approximate the feasible range by adding certain Benders cut constraints sequentially, explicitly determining the feasible range according to the problem structure can reduce the computation time for the iterative procedure. Each iteration can be confined in the feasible range when it is known. Otherwise, some iterative steps have to be used just for refining the approximation of the feasible range.

The detailed procedure of decomposition method I is displayed in Fig. 3. We initialize the iteration by fixing the assignment variables at certain values. The transition times are then fixed at the minimum values. The primal problem consists of a set of separable dynamic optimization problems which can be solved independently. The returned optimal solution to each dynamic optimization problem is used to evaluate the objective function value and update the upper bound value. When the gap is less than the threshold value, the iteration terminates and returns the best solution found during the procedure. Otherwise, the parameters of the master problem are updated according to the optimal value of the dual variables. The master problem is an MINLFP, which should be solved to global optimality to provide a valid lower bound. The detailed solution strategy for this MINLFP is presented in Section 3.3. When it is solved, we record the lower bound value and update the parameters for the dual problem according to the optimal solution to the master problem.

3.3. Global solution to master problem in decomposition method I

The master problem of decomposition method I is an MINLFP which should be solved to global optimality to provide valid Benders cuts. In this section, we develop an efficient global optimization algorithm for the MINLFP master problem that is based on a parametric fractional programming approach (Dinkelbach, 1967; You et al., 2009).

A general expression of a fractional programming problem is given by

$$(Fractional\ P)\qquad \min_{x\in S} Q(x) = \frac{N(x)}{D(x)}$$

where $N(x)$ and $D(x)$ are functions in the numerator and the denominator respectively, S denotes the feasible range. The value of the denominator is assumed to be positive and bounded from zero, i.e. $D(x)\geq\gamma>0$, for a positive number γ.

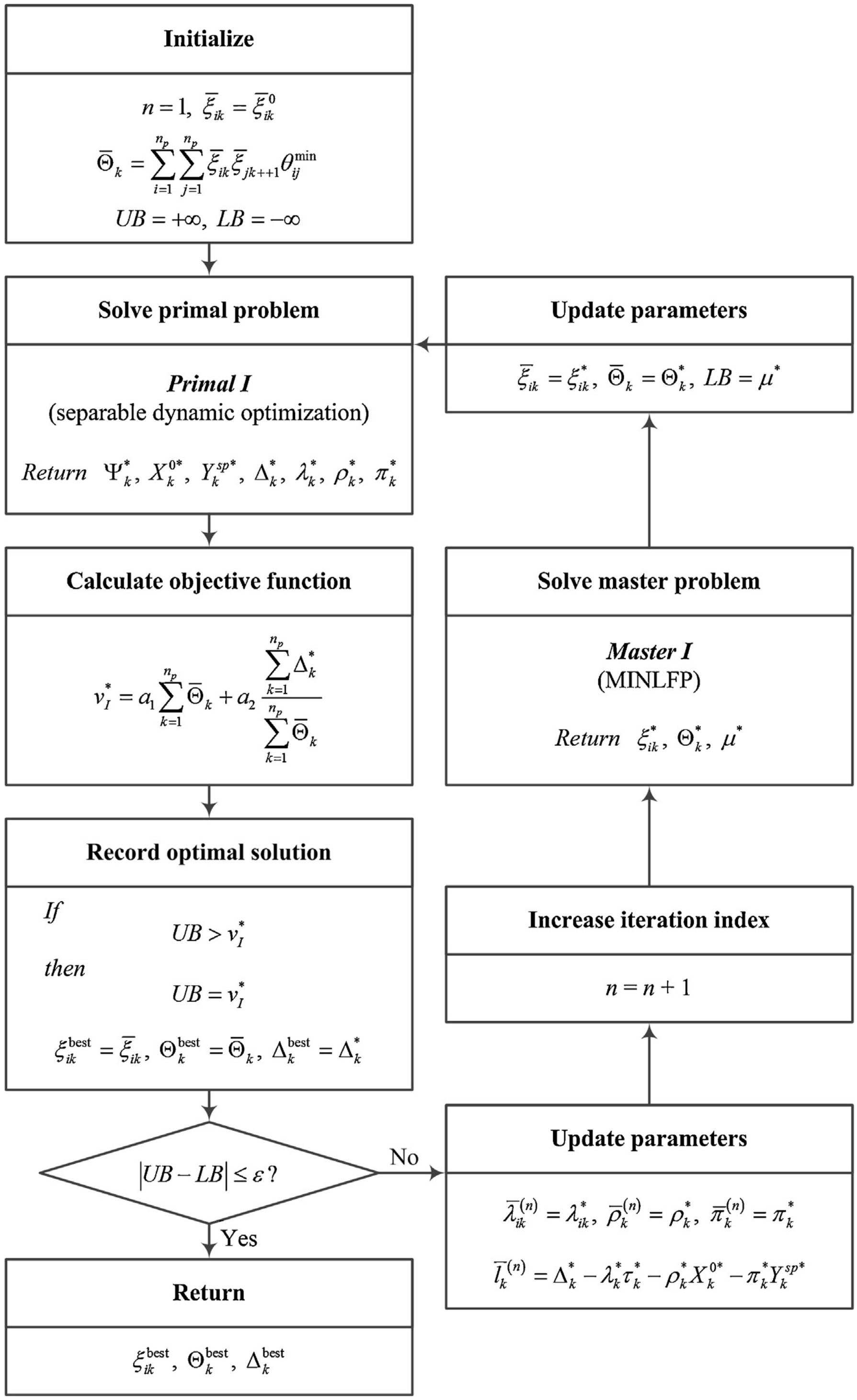

Fig. 3. Procedure of decomposition method I.

The fractional programming problem (*Fractional P*) is equivalent to a parametric programming problem

$$(Parametric\ P)\qquad F(q)=\min_{x\in S} N(x)-qD(x)$$

where q is a parameter. The objective function of the parametric programming problem is the numerator of the original fractional function $N(x)$ minus the denominator $D(x)$ multiplied by the parameter q. The feasible range is the same as the original problem. If we can find the global solution x^* to the parametric programming problem (*Parametric P*) with the parameter q^* such that

$$F(q^*)=0 \tag{26}$$

then x^* is also the global solution to the fractional programming problem (*Fractional P*) (Dinkelbach, 1967; You et al., 2009).

For a fixed parameter q, the parametric programming problem is typically easier to solve than the fractional programming problem. According to the equivalence between the two problems, as long as we can find the global solution to the parametric programming problem with q^* satisfying Eq. (26), the solution is also the global one to the fractional programming problem. To find the solution to the equation, we apply the Newton's method, the parameter value q can be updated in such a way

$$
\begin{aligned}
q_{r+1} &= q_r - \frac{F(q_r)}{-D(x^*)} \\
&= q_r + \frac{N(x^*) - q_r D(x^*)}{D(x^*)} \\
&= \frac{N(x^*)}{D(x^*)}
\end{aligned} \tag{27}
$$

where r denotes the iteration step. x^* is the optimal solution of the parametric programming problem with the parameter value $q = q_r$. $-D(x^*)$ is the subgradient of $F(q)$ a q_r.

Using the fractional programming algorithm, we can solve the master problem in decomposition method I to global optimality. In the master problem (*Master I*), all constraints are linear except the constraint on the transition time (24). Such bilinear term $\xi_{ik}\xi_{jk++1}$ can be linearized by introducing a new variable

$$B_{ijk} = \xi_{ik}\xi_{jk++1} \tag{28}$$

If $B_{ijk} = 1$, then product i is manufactured in time slot k before product j. Then the transition time constraint (24) is equivalent to the following linear constraints

$$\Theta_k \geq \sum_{i=1}^{n_p}\sum_{j=1}^{n_p} B_{ijk}\theta_{ij}^{\min}, \quad \forall k \tag{29}$$

$$B_{ijk} \leq \xi_{ik}, \forall i, j, k \tag{30}$$

$$B_{ijk} \leq \xi_{jk++1}, \forall i, j, k \tag{31}$$

$$B_{ijk} \geq \xi_{ik} + \xi_{jk++1} - 1, \forall i, j, k \tag{32}$$

The corresponding parametric programming problem is

$$(Parametric\,I) \quad F(q) = \min a_1 \left(\sum_{k=1}^{n_p}\Theta_k\right)^2 + a_2\sum_{k=1}^{n_p}\Delta_k - q\sum_{k=1}^{n_p}\Theta_k \tag{33}$$

s.t.
Assignment constrains (2)–(6)
Benders cut constraint (23)
Transition time constraints (29)–(32).

where q is the parameter. The objective function (33) is a quadratic function. Because the parameter q only affects the first-order term, the quadratic function is always convex ($a_1 > 0$) no matter what the value of q is. All constraints of the problem are linear. This convex mixed-integer quadratic programming (MIQCP) problem can be efficiently solved to global optimality using the state-of-the-art branch-and-cut methods implemented by solvers like CPLEX.

Using the fractional programming algorithm, we can find the global solution to the master problem which provides a lower bound for the integrated problem. The detailed procedure for solving the master problem is displayed in Fig. 4, which is embedded into the GBD iterations in Fig. 3. Therefore the fractional programming algorithm can use the information from the primal problem. The information which can be used includes the starting point for the root finding iteration.

In the iteration of the fractional programming algorithm, we need to find the root of the parametric function defined in Eq. (33). We first initialize q at the upper bound value UB returned by the

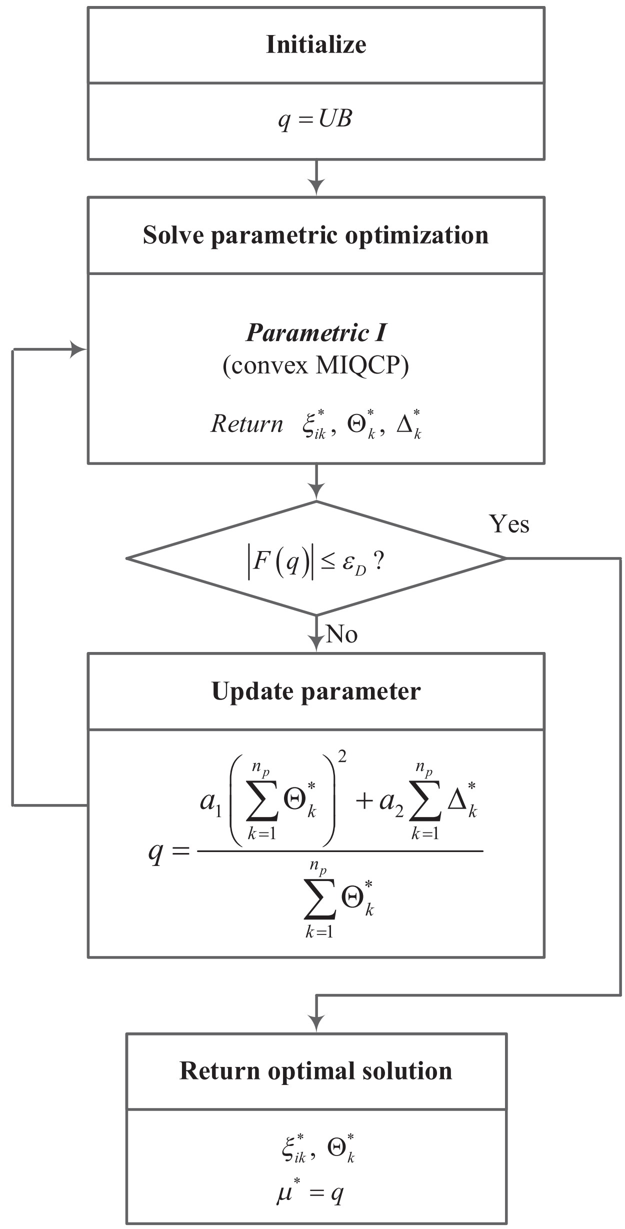

Fig. 4. Fractional programming algorithm for solving the master problem in decomposition method I.

primal problem. Then we solve the parametric optimization problem (*Parametric I*) at q to evaluate the function value $F(q)$. If $F(q)$ is close enough to zero in terms of the threshold value ε_D, then the algorithm stops and returns the global solution ξ_{ik}^* and Θ_k^*. According to the equivalence relation between the parametric problem and the original fractional programming problem, ξ_{ik}^* and Θ_k^* are also the global solution to the master problem (*Master I*).

4. Production sequence formulation and solution method

The time-slot formulation is often used to solve the integrated problem by the simultaneous method (Flores-Tlacuahuac & Grossmann, 2006; Terrazas-Moreno et al., 2007; Zhuge & Ierapetritou, 2012). In this section, we present another new formulation for the integrated problem based on the production sequence model in Section 4.1. The GBD method for the formulation is proposed in Section 4.2 and the global optimization algorithm for the master problem is presented in Section 4.3.

4.1. Formulation based on production sequence model

The main decision in the cyclic scheduling problem is to determine the production sequence. Besides the assignment variables defined in Section 3.1, we can express the production sequence directly by introducing a binary variable β_{ij}. The indices i and j both represent the products. If $\beta_{ij} = 1$, then product i precedes product j. Since the production is carried out cyclically, the last product in a cycle precedes the first product in the subsequent cycle. To distinguish this model from the time-slot assignment model, we call it the production sequence model in this paper. Fig. 5 illustrates the production sequence model.

With the aid of the production sequence variables, β_{ij}, we can express the total transition time and the total transition cost as

$$\sum_{k=1}^{n_p} \Theta_k = \sum_{i=1}^{n_p}\sum_{j=1}^{n_p} \beta_{ij}\theta_{ij} \tag{34}$$

$$\sum_{k=1}^{n_p} \Delta_k = \sum_{i=1}^{n_p}\sum_{j=1}^{n_p} \beta_{ij}\delta_{ij} \tag{35}$$

where θ_{ij} represents the transition time from product i to product j, and δ_{ij} represents the transition cost. Substituting these expressions of the total transition time and the total transition cost into the objective function (1), we reformulate the integrated scheduling and dynamic optimization problem as

$$(Integration\,PS) \quad \min a_1 \sum_{i=1}^{n_p}\sum_{j=1}^{n_p} \beta_{ij}\theta_{ij} + a_2 \frac{\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} \beta_{ij}\delta_{ij}}{\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} \beta_{ij}\theta_{ij}} \tag{36}$$

s.t.

Scheduling model

$$\beta_{ij} \in \left\{0, 1\right\}, \ \forall i, j \tag{37}$$

$$\sum_{j=1}^{N_p} \beta_{ij} = 1, \ \forall i \tag{38}$$

$$\sum_{i=1}^{N_p} \beta_{ij} = 1, \ \forall j \tag{39}$$

$$\beta_{ij} = 0, \ \forall i = j \tag{40}$$

$$\sigma_i - \sigma_j + \left(n_p - 1\right)\beta_{ij} \le n_p - 2, \ \text{for } 2 \le i \ne j \le n_p \tag{41}$$

$$1 \le \sigma_i \le n_p - 1, \ \forall i \ge 2 \tag{42}$$

$$\sigma_{(i=1)} = 1 \tag{43}$$

Dynamic models

$$\frac{dX_{ij}(t)}{dt} = F\left(X_{ij}(t), U_{ij}(t)\right), \ \forall i \ne j \tag{44}$$

$$Y_{ij}(t) = G\left(X_{ij}(t), U_{ij}(t)\right), \ \forall i \ne j \tag{45}$$

$$H\left(X_{ij}(t), U_{ij}(t), Y_{ij}(t)\right) \le 0, \ \forall i \ne j \tag{46}$$

$$X_{ij}(0) = x_i^0, \ \forall i \ne j \tag{47}$$

$$Y_{ij}(t) = y_j^{sp}, \ \forall t \ge \tau_{ij}, \ \forall i \ne j \tag{48}$$

$$\delta_{ij} = \phi_{ij}\left(X_{ij}\left(\tau_{ij}\right), U_{ij}\left(\tau_{ij}\right), Y_{ij}\left(\tau_{ij}\right)\right), \ \forall i \ne j \tag{49}$$

$$\tau_{ij} = \theta_{ij}, \forall i \ne j \tag{50}$$

The scheduling model contains the constraints regarding the binary variables β_{ij} for production sequence. The cyclic scheduling problem based on the production sequence variables is analogous to the traveling salesman problem. In the production scheduling, the products need to be visited one by one in a closed cycle. The optimal cycle is in fact the optimal production sequence we need to determine in the cyclic scheduling. However, the objective function of the cyclic scheduling problem is much more complicated than that of a traveling sales problem. Also, the integrated problem includes nonlinear constraints related to the dynamic models.

The scheduling model contains the standard constraints in the traveling salesman problem. Eqs. (38) and (39) ensure that each product is manufactured exactly once in a cycle. Eq. (40) simply eliminates the infeasible transition from a product to itself. According to this equality, n_p binary variables are fixed and the number of the remaining ones is $n_p \times \left(n_p - 1\right)$. Constraints (41)–(43) aim to eliminate subtours. If a subtour appears, the production is cycled separately inside two or more disjointed groups. We should exclude all cases with subtours. There are various subtour elimination constraints. These used in the integrated problem are the MTZ constraints (Miller, Tucker, & Zemlin, 1960; Yue & You, 2013).

The dynamic model describing the transition process from product i to product j is formulated as constraints (44)–(50). The state and output equations, the path constraint, and the equation defining the transition costs are the same as those in the time-slot model except that the index k is replaced by the indices i and j. However, the initial condition in Eq. (47) and the setpoint values in Eq. (48) are different from those (Eqs. (10) and (11)) in the time-slot model. In the integrated problem based on the product sequence model, the initial condition and the setpoint value are known parameters, i.e. x_i^0 and y_j^{sp}, which are independent on the binary scheduling variables. For the purpose of the decomposition methods, we introduce a duplicated variable of the transition time, denoted by τ_{ij}. The consensus equality (50) guarantees τ_{ij} has the identical value with θ_{ij}.

Comparing with the integrated problem based on the time-slot model, the problem formulated according to the production sequence model has both advantages and disadvantages. Obviously, the production sequence model leads to a much larger integrated problem where, as illustrated in Fig. 5, the number of constituent dynamic models is $n_p \times \left(n_p - 1\right)$ equal to the number of the binary sequence variables. By contrast, the integrated problem based on the time-slot model only includes n_p dynamic models, one for each time slot as illustrated in Fig. 2. The larger scale of the sequence based model makes it more difficult to solve directly.

However, the integrated problem based on the production sequence model has a weak coupling between the scheduling model and the dynamic models. There are no linking equations between the binary scheduling variables and the dynamic models. The constraints of the scheduling model are completely separated from those of the dynamic models. The binary scheduling decision variables interact with the continuous variables in the dynamic models only in the objective function. By contrast, though compact, the integrated problem based on the time-slot assignment model has much stronger interactions between the binary scheduling decision variables and the dynamic models. When the assignment variables change, the initial condition and the setpoint value of at least two dynamic models are changed. Accordingly, all state, input, and output variables in the dynamic models are changed. As will be shown in the next section, the formulation based on the production

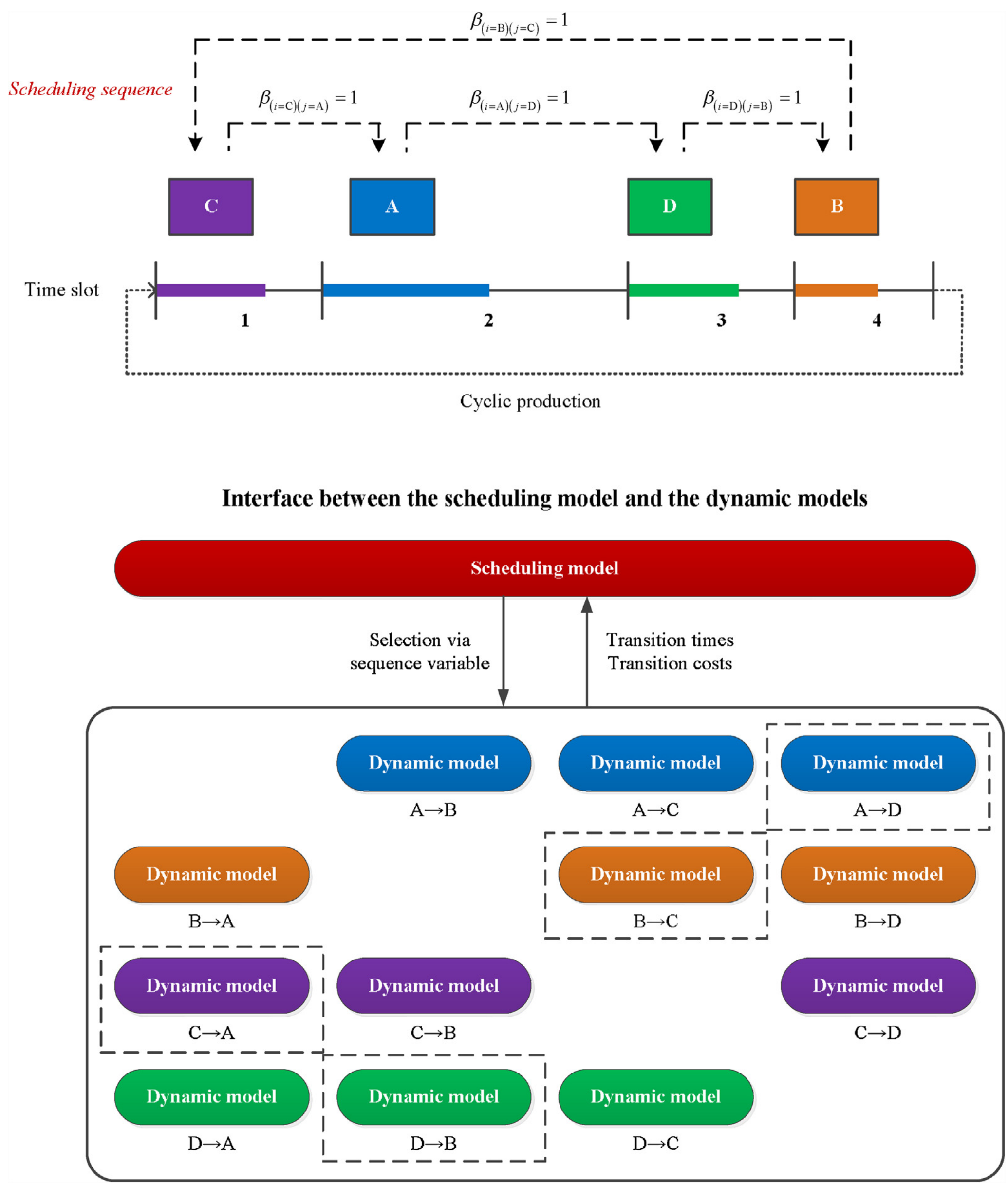

Fig. 5. Illustrative example of the cyclic scheduling problem based on the production sequence model.
The scheduling sequence is determined by the binary variable β_{ij}, which is one when product i precedes product j. The dynamic models in the integrated problem describe the transition period for any possible transition. Using the sequence variable β_{ij}, the scheduling model determines which dynamic models are selected or which transitions should appear in the solution. Then the transition times and the transition costs of the selected dynamic models (or transitions) are returned to the scheduling model. The dashed rectangle indicates the selected dynamic models according to the value of β_{ij} in the top figure.

sequence model can have some advantages for the decomposition purpose due to the separable constraints.

4.2. Method II: decomposition for production sequence formulation

Using the formulation based on the production sequence model, we can obtain another decomposition procedure. In the integrated problem (*Integration PS*) given by Eqs. (36)–(50), the transition cost of a dynamic model is only dependent on the transition time. Thus, we can cast the transition times into the category of the complicating variables along with the binary sequence variables. According to this classification, the projected problem becomes

$$v_{II}\left(\{\bar{\beta}_{ij}\},\{\bar{\theta}_{ij}\}\right) = \min\left(a_1\sum_{i=1}^{n_p}\sum_{j=1}^{n_p}\bar{\beta}_{ij}\bar{\theta}_{ij} + a_2\frac{\sum_{i=1}^{n_p}\sum_{j=1}^{n_p}\bar{\beta}_{ij}\delta_{ij}}{\sum_{i=1}^{n_p}\sum_{j=1}^{n_p}\bar{\beta}_{ij}\bar{\theta}_{ij}}\right) \tag{51}$$

$$= a_1 \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \bar{\beta}_{ij} \bar{\theta}_{ij} + a_2 \frac{\min \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \bar{\beta}_{ij} \delta_{ij}}{\sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \bar{\beta}_{ij} \bar{\theta}_{ij}} \tag{52}$$

$$= a_1 \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \bar{\beta}_{ij} \bar{\theta}_{ij} + a_2 \frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \bar{\beta}_{ij} \min\ \delta_{ij}}{\sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \bar{\beta}_{ij} \bar{\theta}_{ij}} \tag{53}$$

$$= a_1 \sum_{(i,j) \in IJ} \bar{\theta}_{ij} + a_2 \frac{\sum_{(i,j) \in IJ} \min \delta_{ij}}{\sum_{(i,j) \in IJ} \bar{\theta}_{ij}} \tag{54}$$

$$IJ = \left\{ (i,j) \middle| \bar{\beta}_{ij} = 1 \right\} \tag{55}$$

When both $\{\beta_{ij}\}$ and $\{\theta_{ij}\}$ are fixed at $\{\bar{\beta}_{ij}\}$ and $\{\bar{\theta}_{ij}\}$ respectively, the first term in the objective function is fixed. The denominator of the second term is fixed as well. Therefore, we only need to minimize the numerator of the second term and Eq. (51) is equivalent to Eq. (52). Further, the transition cost of a dynamic model is independent on those of other dynamic models so minimizing the sum of the weighted transition cost in Eq. (52) is equivalent to the sum of the minimum transition cost in Eq. (53). In the projected problem, the dynamic models are optimized independently. We can further simplify the expression by removing the terms in the summation with zero value. Given that each $\bar{\beta}_{ij}$ has only two possible values, 0 or 1, only the terms with $\bar{\beta}_{ij} = 1$ need to be evaluated in the projected objective function in Eq. (54). The set IJ in Eq. (55) consists of the paired index (i, j) such that the corresponding elements $\bar{\beta}_{ij}$ is equal to one.

According to the projected objective function, the primal problem is formulated as a set of separated dynamic optimization problems

$$(Primal\ II) \quad \min \delta_{ij}, (i,j) \in IJ$$

s.t.
Dynamic models (44)–(49)

$$\tau_{ij} = \bar{\theta}_{ij} \tag{56}$$

where the original consensus equality (50) is replaced by Eq. (56) where the transition time τ_{ij} is fixed at $\bar{\theta}_{ij}$. We only need to solve the dynamic optimization problems with indices belonging to $IJ = \left\{ (i,j) \middle| \bar{\beta}_{ij} = 1 \right\}$. The size of the set IJ is equal to the number of the products or the time slots, n_p. The primal problem can be solved as easily as that in decomposition method I, even though the two primal problems are derived from different formulations of the integrated problem.

The consensus constraint (56) can provide the dual information by which we can express the projected objective function as

$$\begin{aligned} v_{II}\left(\left\{\bar{\beta}_{ij}\right\}, \left\{\bar{\theta}_{ij}\right\}\right) &= a_1 \sum_{(i,j) \in IJ} \bar{\theta}_{ij} + a_2 \frac{\sum_{(i,j) \in IJ} \max_{\lambda_{ij}} \left[\min \left(\delta_{ij} + \lambda_{ij} \left(\bar{\theta}_{ij} - \tau_{ij} \right) \right) \right]}{\sum_{(i,j) \in IJ} \bar{\theta}_{ij}} \\ &= a_1 \sum_{(i,j) \in IJ} \bar{\theta}_{ij} + a_2 \frac{\sum_{(i,j) \in IJ} \max_{\lambda_{ij}} \left[\min \left(\delta_{ij} - \lambda_{ij} \tau_{ij} \right) + \sum_{(i,j) \in IJ} \lambda_{ij} \bar{\theta}_{ij} \right]}{\sum_{(i,j) \in IJ} \bar{\theta}_{ij}} \end{aligned} \tag{57}$$

where λ_{ij} is the Lagrange multiplier of the consensus constraint. Because the dynamic model is not dependent on the binary sequence variables, we do not need the dual information for those variables. The binary sequence variables are updated directly from the master problem.

The master problem is formulated as

$$(Master\ II) \quad \mu^* = \min a_1 \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \beta_{ij} \theta_{ij} + a_2 \frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \beta_{ij} \delta_{ij}}{\sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \beta_{ij} \theta_{ij}}$$

s.t.
Scheduling model (37)–(43)

$$\delta_{ij} \geq \bar{l}_{ij}^{(m)} + \bar{\lambda}_{ij}^{(m)} \theta_{ij}, \quad \forall (i,j) \in IJ^{(m)}, 1 \leq m \leq n-1 \tag{58}$$

$$\delta_{ij} \geq 0, \quad \forall i, j \tag{59}$$

$$\theta_{ij} \geq \theta_{ij}^{\min}, \quad \forall i, j \tag{60}$$

where m and n denote the iteration numbers. The objective function of the master problem has the same form as that of the integrated problem. The constraints include the scheduling model regarding the binary sequence variables β_{ij}. Constraints (58) represent the Benders cuts. The value of $\bar{l}_{ij}^{(m)}$ is calculated as

$$\bar{l}_{ij}^{(m)} = \delta_{ij}^* - \lambda_{ij}^* \tau_{ij}^* \tag{61}$$

where δ_{ij}^* and τ_{ij}^* belong to the optimal solution to the primal problem. λ_{ij}^* is the optimal dual value regarding Eq. (56) and it is expressed by the parameter $\bar{\lambda}_{ij}^{(m)}$ in the master problem. Inequality (59) provides a lower bound of the transition cost, which should be nonnegative. In inequality (60), the transition time θ_{ij} is bounded from below by the minimum value $\theta_{ij}^{\min}$ which is calculated by solving the dynamic optimization problem (*Minimize TT*).

The detailed procedure of decomposition method II is displayed in Fig. 6. We initialize the iteration by fixing the sequence variables at certain values and fixing the transition times at the minimum values. The primal problem is a set of separable dynamic optimization problems, which are solved to evaluate the objective function value for the upper bound. If the gap is not less than the tolerance, the parameters for the master problem are updated according to the optimal dual values. The master problem is an MINLFP which can be solved by the algorithm presented in Section 4.3. The optimal solution to the master problem is used to update the parameters of the primal problem.

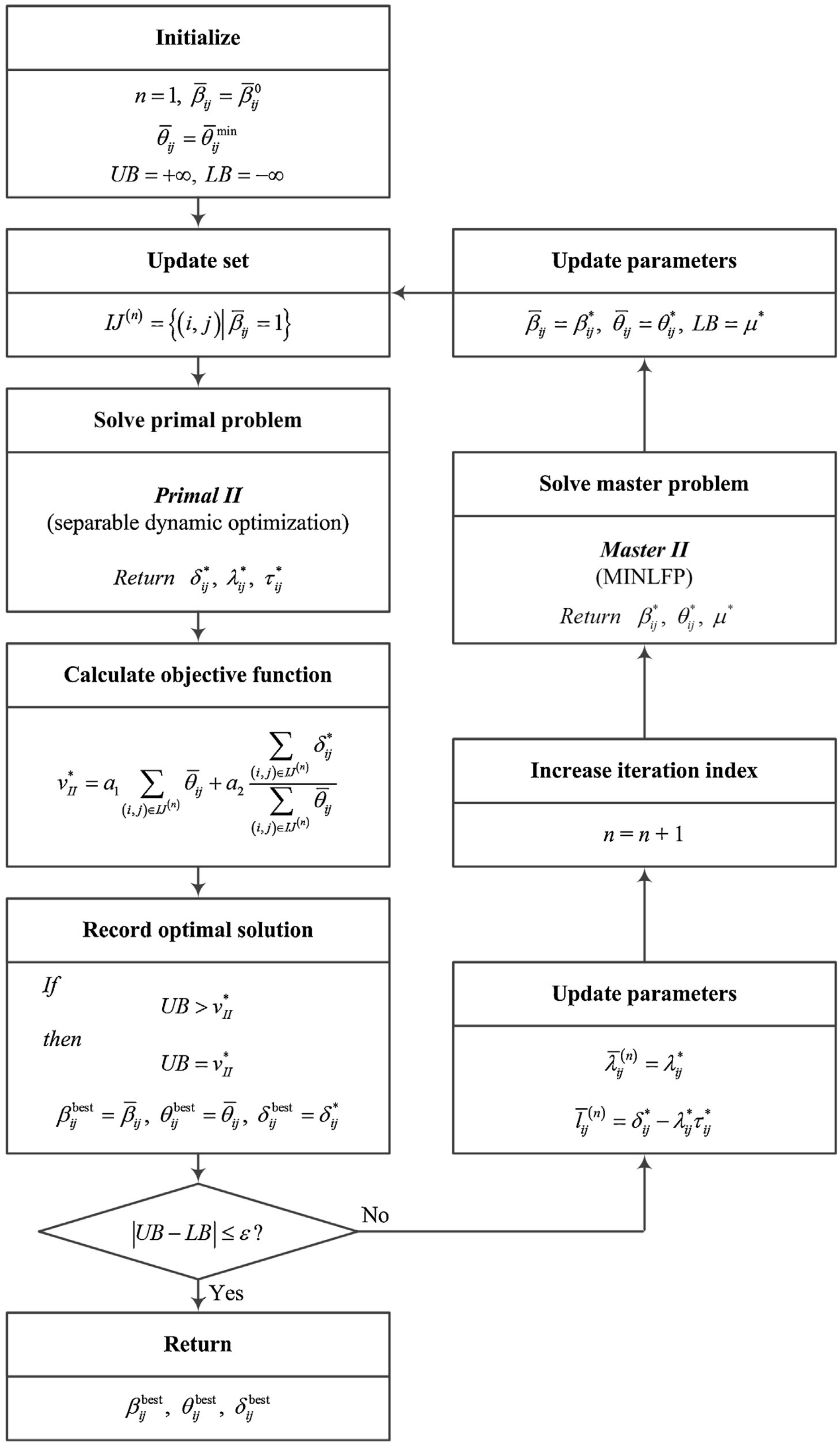

Fig. 6. Procedure of decomposition method II.

4.3. Global solution to master problem in decomposition method II

We can also extend the fractional programming algorithm to solve the master problem in decomposition method II. In the master problem (*Master II*), all constraints have already been linear. All nonlinear terms concentrate on the objective function. Specifically the nonlinear terms are the bilinear terms $\beta_{ij}\theta_{ij}$ and $\beta_{ij}\delta_{ij}$, and the fractional function of the second term. The bilinear terms are products of a binary variable and a continuous variable, which can be linearized. We introduce the continuous variables $BT_{ij} = \beta_{ij}\theta_{ij}$ and $BD_{ij} = \beta_{ij}\delta_{ij}$. Then the constraints representing the bilinear terms are formulated as

$$0 \le BT_{ij} \le \beta_{ij}\theta_{ij}^{\max}, \quad \forall i \ne j \tag{62}$$

$$\theta_{ij} - \left(1-\beta_{ij}\right)\theta_{ij}^{\max} \le BT_{ij} \le \theta_{ij}, \quad \forall i \ne j \tag{63}$$

$$0 \le BD_{ij} \le \beta_{ij}\delta_{ij}^{\max}, \quad \forall i \ne j \tag{64}$$

$$\delta_{ij} - \left(1-\beta_{ij}\right)\delta_{ij}^{\max} \le BD_{ij} \le \delta_{ij}, \quad \forall i \ne j \tag{65}$$

where $\theta_{ij}^{\max}$ and $\delta_{ij}^{\max}$ are the upper bounds of θ_{ij} and δ_{ij}, respectively.

After linearizing the bilinear terms, the master problems become

$$(Parametric\ II) \quad \nu_{II}^{*} = \min a_1 \sum_{i=1}^{n_p}\sum_{j=1}^{n_p} BT_{ij} + a_2 \frac{\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} BD_{ij}}{\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} BT_{ij}}$$

$$= \min \frac{a_1\left(\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} BT_{ij}\right)^2 + a_2 \sum_{i=1}^{n_p}\sum_{j=1}^{n_p} BD_{ij}}{\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} BT_{ij}} \tag{66}$$

s.t.

Linear constraints (37)–(43), (58)–(60), and (62)–(65).

In the fractional objective function, the numerator is a convex quadratic function and the denominator is a linear function. All constraints are linear. This fractional programming problem is equivalent to a parametric mixed-integer quadratic programming problem

$$F(q) = \min a_1\left(\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} BT_{ij}\right)^2 + a_2 \sum_{i=1}^{n_p}\sum_{j=1}^{n_p} BD_{ij} - q\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} BT_{ij} \tag{67}$$

s.t.

Linear constraints (37)–(43), (58)–(60), and (62)–(65)

where q is the parameter. For a fixed parameter q, the parametric programming problem is easier to solve than the fractional programming problem. The parametric programming problem is convex ($a_1 > 0$) no matter what the value of q is. The convexity implies that any local solution is the global one.

The detailed procedure for solving the master problem is displayed in Fig. 7, which is embedded into the GBD iterations in Fig. 6. We initialize the parameter q at the upper bound UB returned by the primal problem. Then the convex quadratic programming problem (*Parametric II*) is solved with the parameter value q to evaluate the parametric function $F(q)$. If $F(q)$ is close to zero in terms of the threshold value ε_D, then the algorithm stops and returns the global solution β_{ij}^{*} and θ_{ij}^{*}, which are also the global solution to the master problem (*Master II*). The optimal function value of the master problem is equal to the value of q, i.e. $\mu^{*} = q$. Otherwise, the parameter value q is updated.

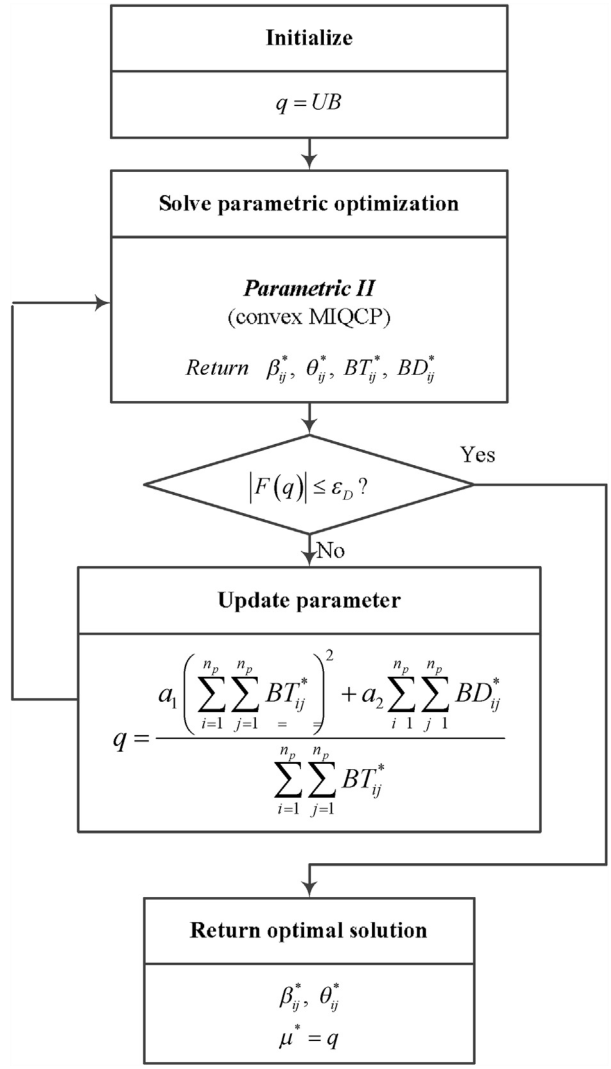

Fig. 7. Fractional programming algorithm for solving the master problem in decomposition method II.

5. Case studies

We demonstrate the GBD methods by a case study of a polymerization process. For the sake of comparison, the integrated problem is solved first by the simultaneous method. The simultaneous method solves the integrated problem (*Integration TS*) in Section 3.1 by discretizing the differential equations via the collocation approach (Biegler, 2007; Cuthrell & Biegler, 1987). The resulting MINLP problem is then solved by SBB (Flores-Tlacuahuac & Grossmann, 2006; Nie et al., 2012; Terrazas-Moreno et al., 2007; Zhuge & Ierapetritou, 2012). Then, the two decomposition methods are applied and the results are compared with those returned by the simultaneous method.

For consistency, we apply the collocation method to solve all dynamic optimization problems for the simultaneous method and the decomposition methods. The settings of the dynamic optimization problems, e.g. the number of the finite elements, the number of collocation points, the lower or upper bound of each variable, are identical for all methods. The dynamic optimization problems are also initialized in the same way.

In this work, all optimization problems are modeled in GAMS 24.0.1 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.

Table 1
Data of the dynamic model of the MMA polymerization process.

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	$2.50 \times 10^6\ m^3/kmol\,h$
k_{T_d}	Termination by disproportionation rate constant	$1.09 \times 10^{11}\ m^3/kmol\,h$
k_{T_c}	Termination by coupling rate constant	$1.33 \times 10^{10}\ m^3/kmol\,h$
$C_{I_{in}}$	Inlet initiator concentration	$8.00\ kmol/m^3$
$C_{m_{in}}$	Inlet monomer concentration	$6.00\ kmol/m^3$
k_{fm}	Chain transfer to monomer rate constant	$2.45 \times 10^3\ m^3/kmol\,h$
k_I	Initiation rate constant	$1.02 \times 10^{-1}\ h^{-1}$
M_m	Molecular weight of monomer	100.12 kg/kmol
c^r	Unit cost of initiator	$10^5\ m.u./m^3$

Table 2
Steady-state values of 12 products in MMA process.

Product	A	B	C	D	E	F
Y_i^{ss} (10^4 kg/kmol)	1.5	1.7	1.9	2.1	2.3	2.5
Product	G	H	I	J	K	L
Y_i^{ss} (10^4 kg/kmol)	2.7	2.9	3.1	3.3	3.5	3.7

5.1. Integrated problem

This example is a methyl methacrylate (MMA) polymerization process. The set of differential equations (Mahadevan, Doyle, & Allcock, 2002; Terrazas-Moreno, Flores-Tlacuahuac, & Grossmann, 2008) is summarized below and the parameters in the model are listed in Table 1.

Differential equations

$$\begin{aligned}
\frac{dC_m(t)}{dt} &= -\left(k_p + k_{fm}\right)\sqrt{\frac{2f^*k_I}{k_{T_d}+k_{T_c}}}C_m(t)\sqrt{C_I(t)} + \frac{F\left(C_{m_{in}}(t) - C_m(t)\right)}{V} \\
\frac{dC_I(t)}{dt} &= -k_I C_I(t) + \frac{F_I(t)C_{I_{in}}(t) - FC_I(t)}{V} \\
\frac{dD_0(t)}{dt} &= \left(0.5k_{T_c} + k_{T_d}\right)\frac{2f^*k_I}{k_{T_d}+k_{T_c}}C_I(t) + k_{fm}\sqrt{\frac{2f^*k_I}{k_{T_d}+k_{T_c}}}C_m(t)\sqrt{C_I(t)} - \frac{FD_0(t)}{V} \\
\frac{dD_1(t)}{dt} &= M_m\left(k_p + k_{fm}\right)\sqrt{\frac{2f^*k_I}{k_{T_d}+k_{T_c}}}C_m(t)\sqrt{C_I(t)} - \frac{FD_1(t)}{V}
\end{aligned} \tag{68}$$

Input

$$U(t) = F_I(t)$$

Output

$$Y(t) = \frac{D_1(t)}{D_0(t)}$$

Transitioncost

$$\Delta = c^r \int_0^{\Theta} U(t)\,dt$$

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 manipulated input $U(t)$ is the flow rate of the initiator. The output $Y(t)$ represents the molecular weight of each product. The transition cost Δ is the consumption of the initiator during the transition period.

There are 12 products manufactured in the process: A, B, C, D, . . ., K, L. Each product has a specific steady-state value of the molecular weight, Y_i^{ss}, listed in Table 2. 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 production rate of each product $g_i = 10\ m^3/h$. The demand rate is $d_i = 0.8\ m^3/h$ and the unit price of the initiator is $c^r = 10\ m.u./m^3\,h$.

Table 3
Model and solution statistics of the simultaneous and the decomposition methods for 8 products.

Method		Simultaneous	Decomposition	
			I	II
Complicating variables		–	Assignment $\{\xi_{ik}\}$ Transition time $\{\Theta_k\}$	Sequence $\{\beta_{ij}\}$ Transition time $\{\theta_{ij}\}$
Formulation		*Integration TS*	*Integration TS*	*Integration PS*
Number of iterations		–	4	13
Total CPU (s)		14,354.7	19.9	49.3
Objective (m.u./h)		225.0[b]	225.0	225.0
Optimal sequence		A → B → C → D → E → F → G → H	A → B → C → D → E → F → G → H	A → B → C → D → E → F → G → H
Gap (%)		1.0	1.0	1.0
Primal problem	Type	–	Separable NLP	Separable NLP
	Equations	–	8 × 4087	8 × 4087
	Variables	–	8 × 4011	8 × 4011
	Solver	–	CONOPT 3	CONOPT 3
	CPU (s)	–	7.2	38.4
Integrated/Master problem[a]	Type	MINLP	MINLFP	MINLFP
	Equations	32,708	164 (first iteration) 188 (last iteration)	454 (first iteration) 550 (last iteration)
	Variables	32,156 (all) 63 (binary)	596 (all) 63 (binary)	331 (all) 56 (binary)
	Algorithm	SBB	Fractional programming + CPLEX	Fractional programming + CPLEX
	CPU (s)	14,354.7	12.7	10.9
	Gap (%)	1.0	0	0

[a] For the simultaneous method, the statistics correspond to the integrated problem. For the decomposition methods, the statistics correspond to the master problem.
[b] There are 1344 nodes explored and the optimal solution is found at the 1344th node.

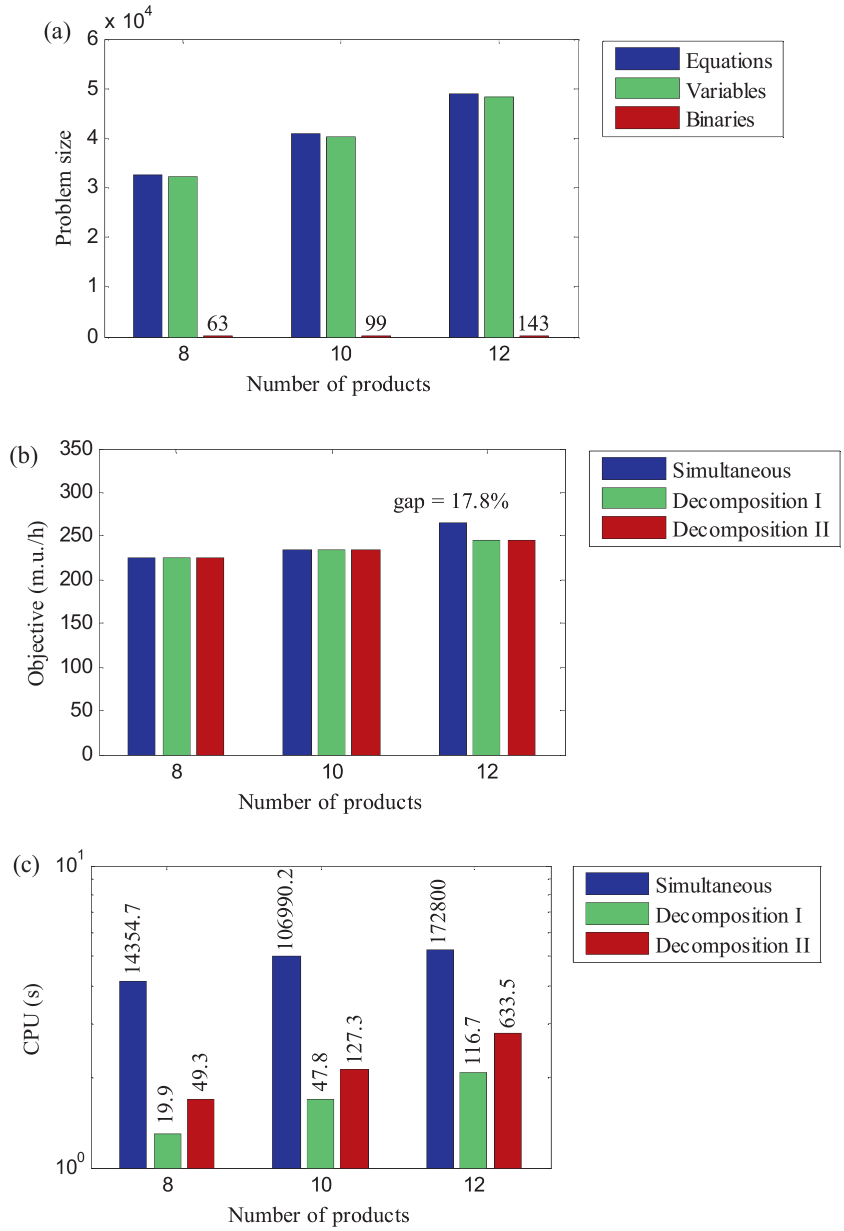

Fig. 8. Results for different numbers of products. (a) The model statistics. (b) The objective function values returned by the simultaneous method and the decomposition methods I and II. (The limit of the computational time is 172,800 s (48 h). The simultaneous method does not solve the problem to the 1% optimality gap for the 12-product problem. The gap when the solver terminates after 48 h is shown in the figure.) (c) The computational times.

5.2. Comparison results

The polymerization process is a highly nonlinear system. Many square-root and bilinear terms are present in the differential equations. The strong nonlinearity demands a large number of finite elements in the collocation method for an accurate approximation of the solution to the differential equations. We choose 100 finite elements to discretize the differential equations.

5.2.1. Comparison for three problems

Even for a single CSTR, the integrated problem is too complicated for the simultaneous method to solve. There are 49,060 equations, 48,280 variables, and 143 binaries in the MINLP problem. Within the computational limit of 172,800 s (48 h), the simultaneous method cannot solve the integrated problem to the 1% optimality gap. The terminated gap by SBB is as large as 17.8% after 48 h. To make a thorough comparison, we also derive two simpler problems by reducing the number of products. There are three problems solved. The first one is a 8-product problem with products from A to H in Table 2. The second one is a 10-product problem with products from A to J. The third one is the entire 12-product problem with products from A to L.

The problem size for each problem is summarized in Fig. 8(a). The objective function values returned by the simultaneous method and the decomposition methods are shown in Fig. 8(b). For the 8-product problem and the 10-product problem, the

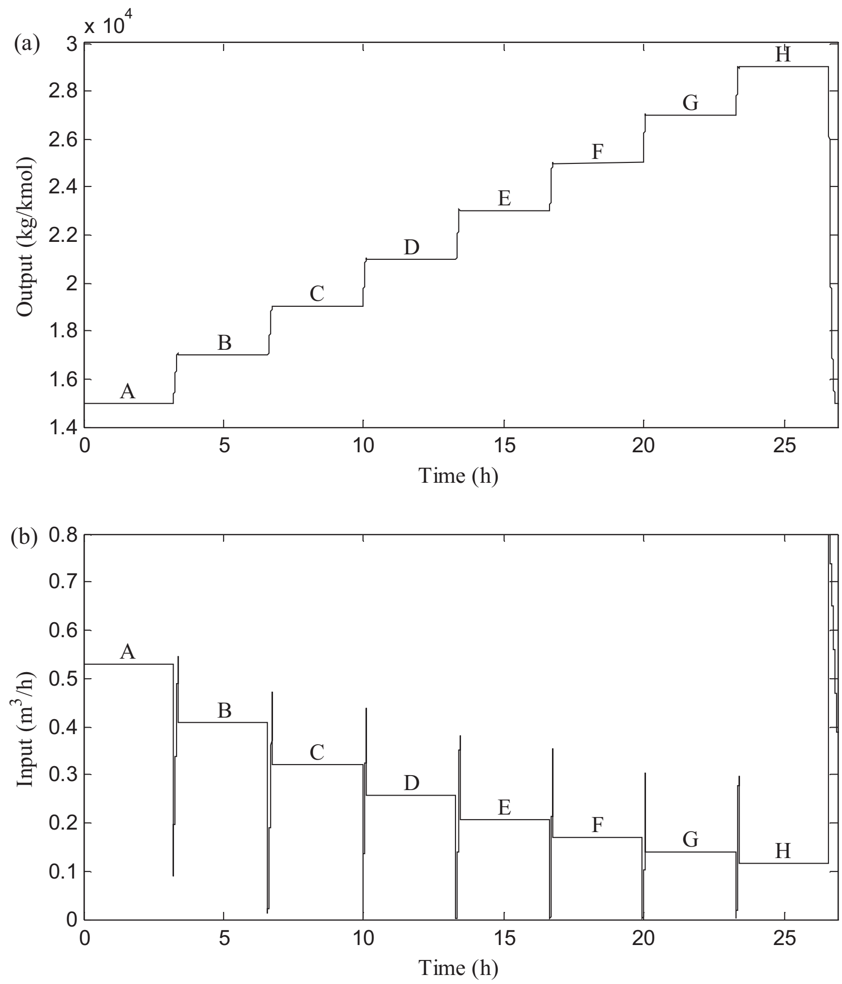

Fig. 9. Dynamic profiles by solving the 8-product problem: (a) output and (b) input.

decomposition methods return the same optimal value as the simultaneous method. The minimum cost for the 8-product problem is 225.0 m.u./h and the one for the 10-product problem is 235.1 m.u./h. The integrated problems are solved to the 1% optimality gap. However, for the 12-product problem, the simultaneous method only returns a suboptimal solution within 48 h. The cost returned by the simultaneous method is 266.0 m.u./h, which is 8.2% larger than 245.9 m.u./h by the decomposition methods. The computational times are shown in Fig. 8(c). For the three problems, decomposition method I has the shortest computational times, which are 0.14%, 0.04%, and 0.07% of those for the simultaneous method. Though decomposition method II is slower than decomposition method I, it requires much shorter computational times than the simultaneous method. The computational times are 0.34%, 0.12%, 0.37% of those for the simultaneous method.

5.2.2. Results of 8-product problem

To obtain more details about the decomposition methods, we investigate the 8-product problem. The model and solution statistics are summarized in Table 3. This is a large-scale MINLP problem. Though the integrated problem contains only 63 binary variables, it includes more than 30,000 equations and continuous variables. We initialize the problem with the production sequence A → G → F → H → E → C → B → D. The variables regarding the dynamic models are initialized by the steady-state values according to the production sequence. The optimal sequence returned by the simultaneous method is A → B → C → D → E → F → G → H. The dynamic profiles of the input and output variables are displayed in Fig. 9.

Due to the large-scale, the MINLP solver SBB needs nearly 4 h to obtain the optimal solution with the 1% gap. The computational time can be significantly reduced by the decomposition methods. The model and solution statistics for the decomposition methods are listed in Table 3 in parallel with those of the simultaneous method. For decomposition method I, the total computational time is reduced to 19.9 s (nearly by three orders of magnitudes). Only four iterations are required to obtain the same objective function value as the simultaneous method. Decomposition method II uses a longer computational time of 49.3 s than decomposition method I. The number of iterations increases to 13. However, comparing with the simultaneous method, the computational time is reduced to less than 1%. The convergence procedures of both decomposition methods are shown in Fig. 10.

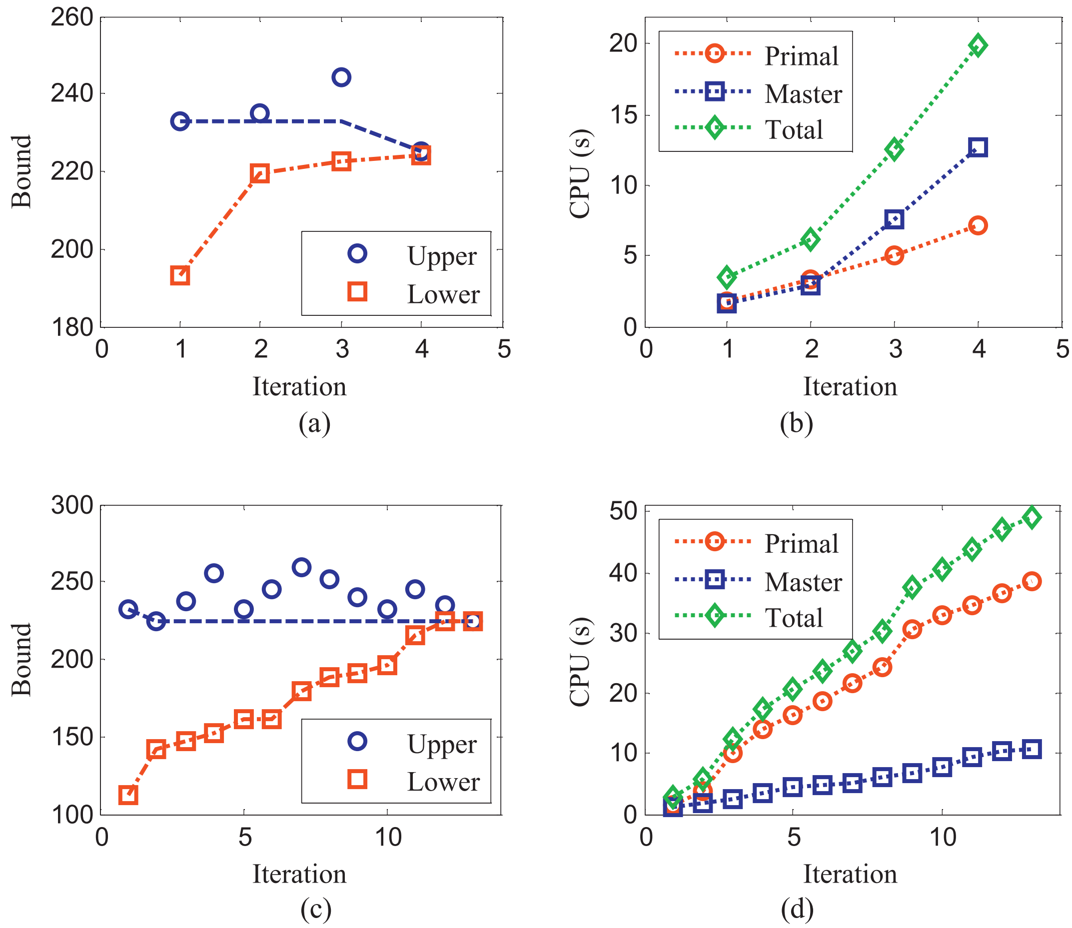

Fig. 10. Iterative results of the decomposition methods: (a) bounds for decomposition method I, (b) computational times for decomposition method I, (c) bounds for decomposition method II, (d) computational times for decomposition method II.

6. Conclusion

Integrated production scheduling and dynamic optimization for a single multi-product CSTR is a complicated MIDO problem. To simplify the computational complexity, we proposed two decomposition methods based on the GBD framework. The two decomposition methods were applied to two formulations of the integrated problem. Decomposition method I was applied to the time-slot formulation. It decomposed the integrated problem into a primal problem consisting of separable dynamic optimization problems and a master problem of MINLFP. Decomposition method II decomposed the integrated problem into two subproblems like decomposition method I but it was applied to the production sequence formulation. The master problems of the two decomposition methods were globally optimized by the efficient fractional programming algorithms.

The two decomposition methods were demonstrated by a polymerization process. Comparisons with the simultaneous method were made for three problems. For the 8-product problem and the 10-product problems, the decomposition methods returned the same minimum cost as the simultaneous method. However, the computational times were reduced by more than two orders of magnitudes. For the entire 12-product problem, decomposition method I solved the integrated problem in about 2 min while decomposition method II required about 6 min. For such a complicated problem, the simultaneous method only returned a suboptimal value within 48 h.

The proposed methods can be extended to multiple CSTRs by introducing binary allocation variables. The decomposition structure will be similar to those for a single CSTR. The master problem is an MINLFP and the primal problem consists of separable dynamic optimization problems. When the production sequence variables (or the time-slot assignment variables), the unit allocation variables, and the transition times are fixed, the dynamic optimization problems for each time slot in a unit can be solved independently.

References

Ascher, U. M., Ruuth, S. J., & Spiteri, R. J. (1997). Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, *25*, 151–167.

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.

Barton, P. I., Allgor, R. J., Feehery, W. F., & Galan, S. (1998). Dynamic optimization in a discontinuous world. *Industrial & Engineering Chemistry Research*, *37*, 966–981.

Bassett, M. H., Dave, P., Doyle, F. J., Kudva, G. K., Pekny, J. F., Reklaitis, G. V., et al. (1996). Perspectives on model based integration of process operations. *Computers & Chemical Engineering*, *20*, 821–844.

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

Chatzidoukas, C., Pistikopoulos, S., & Kiparissides, C. (2009). A hierarchical optimization approach to optimal production scheduling in an industrial continuous olefin polymerization reactor. *Macromolecular Reaction Engineering*, *3*, 36–46.

Chu, Y., & You, F. (2012). Integration of scheduling and control with online closed-loop implementation: Fast computational strategy and large-scale global optimization algorithm. *Computers & Chemical Engineering*, *47*, 248–268.

Chu, Y., & You, F. (2013a). Integrated scheduling and dynamic optimization of sequential batch processes with online implementation. *AIChE Journal*, *59*, 2379–2406.

Chu, Y., & You, F. (2013b). Integrated scheduling and dynamic optimization of complex batch processes with general network structure using a generalized benders decomposition. *Industrial & Engineering Chemistry Research*, *52*, 7867–7885.

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*, 492–498.

Engell, S., & Harjunkoski, I. (2012). Optimal operation: Scheduling, advanced control and their integration. *Computers & Chemical Engineering*, *47*, 121–133.

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

Floudas, C. A. (1995). *Nonlinear and mixed-integer optimization: Fundamentals and applications*. New York, USA: Oxford University Press.

Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of Optimization Theory and Applications*, *10*, 237–260.

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.

Klatt, K. U., & Marquardt, W. (2009). Perspectives for process systems engineering-personal views from academia and industry. *Computers & Chemical Engineering*, *33*, 536–550.

Li, X., Chen, Y., & Barton, P. I. (2012). Nonconvex generalized Benders decomposition with piecewise convex relaxations for global optimization of integrated process design and operation problems. *Industrial & Engineering Chemistry Research*, *51*, 7287–7299.

Mahadevan, R., Doyle, F. J., & Allcock, A. C. (2002). Control-relevant scheduling of polymer grade transitions. *AIChE Journal*, *48*, 1754–1764.

Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the Association for Computing Machinery*, *7*, 326–329.

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.

Mohideen, M. J., Perkins, J. D., & Pistikopoulos, E. N. (1996). Optimal design of dynamic systems under uncertainty. *AIChE Journal*, *42*, 2251–2272.

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*, *58*, 3416–3432.

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.

Oldenburg, J., Marquardt, W., Heinz, D., & Leineweber, D. B. (2003). Mixed-logic dynamic optimization applied to batch distillation process design. *AIChE Journal*, *49*, 2900–2917.

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.

Sahinidis, N. V., & Grossmann, I. E. (1991). Convergence properties of generalized benders decomposition. *Computers & Chemical Engineering*, *15*, 481–491.

Sakizlis, V., Perkins, J. D., & Pistikopoulos, E. N. (2004). Recent advances in optimization-based simultaneous process and control design. *Computers & Chemical Engineering*, *28*, 2069–2086.

Terrazas-Moreno, S., Flores-Tlacuahuac, A., & Grossmann, I. E. (2007). Simultaneous cyclic scheduling and optimal control of polymerization reactors. *AIChE Journal*, *53*, 2301–2315.

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.

Turkay, M., & Grossmann, I. E. (1996). Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers & Chemical Engineering*, *20*, 959–978.

Wassick, J. M., Agarwal, A., Akiya, N., Ferrio, J., Bury, S., & You, F. (2012). Addressing the operational challenges in the development, manufacture, and supply of advanced materials and performance products. *Computers & Chemical Engineering*, *47*, 157–169.

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.

Yue, D., & You, F. (2013). Planning and scheduling of flexible process networks under uncertainty with stochastic inventory: MINLP models and algorithm. *AIChE Journal*, *59*, 1511–1532.

Zhuge, J. J., & Ierapetritou, M. G. (2012). Integration of scheduling and control with closed loop implementation. *Industrial & Engineering Chemistry Research*, *51*, 8550–8565.