Lagrangean Heuristic for the Scheduling and Control of Polymerization Reactors

Sebastian Terrazas-Moreno and Antonio Flores-Tlacuahuac
Departamento de Ingeniería y Ciencias Químicas, Universidad Iberoamericana, Prolongación Paseo de la Reforma 880, México D.F., 01210, México

Ignacio E. Grossmann
Dept. of Chemical Engineering, Carnegie-Mellon University, 5000 Forbes Av., Pittsburgh 15213, PA

DOI 10.1002/aic.11343
Published online November 20, 2007 in Wiley InterScience (www.interscience.wiley.com).

A decomposition technique is applied to address the simultaneous scheduling and optimal control problem of multigrade polymerization reactors. The simultaneous scheduling and control (SSC) problem is reformulated using Lagrangean decomposition as presented by Guignard and Kim. The resulting model is decomposed into scheduling and control subproblems, and solved using a heuristic approach used before by Van den Heever et al. in a different kind of problem. The methodology is tested using a methyl methacrylate (MMA) polymerization system, and the high impact polystyrene (HIPS) polymerization system, with one continuous stirred-tank reactor (CSTR), and with a complete HIPS polymerization plant composed of a train of seven CSTRs. In these case studies, different polymer grades are produced using the same equipment in a cyclic schedule. The results of the heuristic decomposition technique are compared against those obtained by solving the problem without decomposition, whenever both solutions were available. The presence of a duality gap for the decomposed solution is observed as expected when integer variables and other nonconvexities are present. Computational times in the first two examples were lower for the decomposition heuristic than for the direct solution in full space, and the optimal solutions found were slightly better. The example related to the full scale HIPS plant was only solvable using the decomposition heuristic. © 2007 American Institute of Chemical Engineers *AIChE J,* 54: 163–182, 2008

Keywords: optimization, process control, polymerization

Introduction

The simultaneous scheduling and control (SSC) problem involves determining the best sequencing of products for a manufacturing operation, while at the same time determining the optimal dynamic trajectories during product transitions. The simultaneous approach has shown to result in improved operation of certain chemical engineering systems compared to the traditional sequential approach.[3,4] The sequential approach involves the determination of the production schedule and the optimal product transitions in two different steps. The SSC problem in polymerization systems has been addressed in a number of ways.[4–10] Some works compare different product transitions for a polymerization CSTR.[5,6] In these works robust control theory is employed as screening tools and heuristics for determining favorable transitions, and a preferred schedule is proposed based on those tools. Prata et al.[7] propose a simultaneous scheduling and control formulation in which the dynamic optimization part is solved using multiple shooting techniques. This formulation was tested

Correspondence concerning this article should be addressed to A. Flores-Tlacuahuac at antonio.flores@uia.mx.

using a continuous polymerization reaction system under different demand patterns (with and without due dates). Nystrom et al.[8,9] report a solution methodology for the SSC problem in which the original mixed integer dynamic optimization (MIDO) formulation is decoupled into a master scheduling problem, and a primal control problem. The two problems are solved separately in an iterative manner, while information regarding key high-level parameters is exchanged between them. The key high-level parameters exchanged include transition and production times obtained from the primal problem, and are included in the solution of the master problem. This methodology completely decouples the sequencing and the control problem. The solution to be proposed in this article is different from the last two references discussed,[8,9] since the two subproblems in this work, share a common set of dualized constraints. Therefore, the SSC problem is decomposed rather than decoupled. In other works.[4,10] assuming cyclic schedules, the SSC problem in multiproduct CSTRs turns out to be a MIDO problem, which is solved using the simultaneous dynamic optimization (SDO) approach.[11]

The objective of this article is to solve the simultaneous scheduling and control (SSC) problem, based on our previous formulation,[4] by exploiting its decomposable nature through a Lagrangean decomposition technique.[1] The reformulated SSC model is solved using a decomposition technique using a heuristic iterative procedure known to be useful for MINLP problems. In this procedure a set of upper bounds for the maximization problem is obtained through the rigorous solution of the decomposed model, while lower bounds are obtained by solving an NLP in which the binary variables are fixed using heuristics. Van den Heever et al.[2] found that this technique can greatly reduce the time spent solving large MINLPs in the area of oil field planning. The same work also shows that if the problem size is large enough, the decomposition heuristic allows the solution of otherwise unsolvable problems.

The idea of applying decomposition techniques to large-scale scheduling problems has been presented before. A good example is the work by de Matta,[12] where a Lagrangean decomposition is used to solve a singleline multiproduct scheduling problem. This work is similar to ours in that it minimizes total inventory and changeover costs over a certain number of production periods. However, de Matta does not consider transition dynamics, changeover time is assumed to be negligible, and changeover cost is not dependent on product sequence. Another example of the use of decomposition techniques for scheduling problems is presented by Wu et al.[13] In this work, a number of different decomposition approaches are used in a reaction-separation sequence represented by a state task network. One of these approaches is the Lagrangean decomposition, which once again is coupled with a heuristic technique in order to generate a sequence of upper and lower bounds. This work does not take into consideration the dynamics of transitions, which is the fundamental feature of our SSC formulation. As a follow up to that work,[13] Wu et al.[14] worked on an improved approach for updating the Lagrangean multipliers, and apply their findings to a scheduling problem. Their formulation still does not include process dynamics, but it does take care of the disadvantages of the most common methods for updating the Lagrangean multipliers, although it carries added complexity.

In this article Lagrangean decomposition of the SSC problem in polymerization reactors is proposed. The size of the resulting subproblems, the computational effort and the optimal solution are analyzed. The performance of this technique is compared against the direct solution obtained in the full space, whenever both were obtained. In the remainder of this article the term "direct solution" refers to the solution obtained in the full solution space without decomposing the SSC problem.

In order to place our decomposition strategy in the context of alternative solutions, other than the direct solution of the SSC problem, the last and largest example problem presented in this article is solved using a sequential approach. In this approach, the dynamic optimization problem that corresponds to each possible dynamic transition in the system is solved separately, and the resulting transition costs and durations are fed into the scheduling problem as fixed parameters. The sequential solution can be thought of as a heuristic or even intuitive decomposition strategy, and it represents the most common way to deal with the dynamic elements of transitions in most scheduling and control problems. Using the sequential approach as a reference is reasonable since it represents the starting point from which all benefits of using a simultaneous approach should be quantified.

Problem Definition

Previously, we proposed a simultaneous scheduling and control (SSC) formulation for polymerization reactors.[4] In this article the same problem definition for the SSC problem will be used. The objective of the SSC problem is to determine the optimal production campaign in a cyclic polymerization operation, where the production of different polymer grades is carried out using the same equipment. Different polymer grades are obtained from the same raw material using different process conditions, which result in different end product characteristics. The optimal cycle is described by the following decision variables: (1) grade manufacturing sequence, (2) variables involved in dynamic transitions, (3) cycle duration, and (4) amounts of grades produced. Certain assumptions are made in order to obtain the optimal solution: (a) demand, inventory costs and raw material costs are deterministic parameters, (b) all polymer produced is sold; there is no upper bound on the demand, (c) all grades are produced only once during the production cycle, (d) once a grade has been produced it is stored and depleted until the end of the cycle, and (e) profit is defined as product sales minus inventory holding costs minus transition costs divided by total cycle time (hourly profit).

The scheduling and the control problems in the SSC formulation share a limited number of variables. In the formulation used in this article such variables are binary sequencing variables, transitions duration, and cycle duration. The Lagrangean decomposition technique exploits this characteristic and reformulates the SSC problem to obtain two separable subproblems. This article is concerned with the solution of the decomposed SSC problem in polymerization reactors, and the comparison of this solution against that obtained by a direct solution.

Scheduling and Control MIDO Formulation

For convenience to the reader, we present the model in concise form. For a detailed description of the model see.[4] The manufacturing operation relevant to this work is carried out in production cycles. The cycle time is divided into a series of slots. Within each slot two operations are carried out: (a) the production period during which a given product is manufactured around steady-state conditions, and (b) the transition period during which dynamic transitions between two products take place. It is assumed that only one product can be produced in a slot, and that each product is produced only once within each production wheel. Also, once a production wheel is completed, new identical cycles are executed indefinitely. The notation is listed in detail in the Appendix.

Objective function.

$$\max \quad \left\{ \sum_{i=1}^{N_p} \frac{C_i^p W_i}{T_c} - \sum_{i=1}^{N_p} \frac{C_i^s (G_i - W_i/T_c)\Theta_i}{2} - \left[\sum_{k=1}^{N_s} \sum_{f=1}^{N_f e} h_{fk} \theta_k^t Q_{max}^m \sum_{c=1}^{N_{cp}} u_{fck}^m \gamma_c \right] \frac{C^r}{T_c} - \left[\sum_{k=1}^{N_s} \sum_{f=1}^{N_f e} h_{fk} \theta_k^t Q_{max}^I \sum_{c=1}^{N_{cp}} u_{fck}^I \gamma_c \right] \frac{C^I}{T_c} \right\} \quad (1)$$

The total process profit is given by the income of the manufactured products minus the sum of the inventory costs and the product transition costs. All terms are divided by the cyclic time (T_c) so that the objective function value corresponds to the cyclic hourly profit.

In order to clarify the SSC MIDO problem formulation, the constraints have been divided into two parts: (1) scheduling part, and (2) dynamic optimization part.

1. Scheduling.

a) Product assignment

$$\sum_{k=1}^{N_s} y_{ik} = 1, \ \forall i \quad (2a)$$

$$\sum_{i=1}^{N_p} y_{ik} = 1, \ \forall k \quad (2b)$$

where y_{ik} is the Binary variable to denote if product i is produced at slot k

b) Amounts manufactured

$$W_i \geq D_i T_c, \ \forall i \quad (3a)$$

$$W_i = G_i \Theta_i, \ \forall i \quad (3b)$$

$$G_i = F_i^o C_m^f MW_{monomer}, \ \forall i \quad (3c)$$

c) Processing times

$$\theta_{ik} \leq \theta^{max} y_{ik}, \ \forall i,k \quad (4a)$$

$$\Theta_i = \sum_{k=1}^{N_s} \theta_{ik}, \ \forall i \quad (4b)$$

$$p_k = \sum_{i=1}^{N_p} \theta_{ik}, \ \forall k \quad (4c)$$

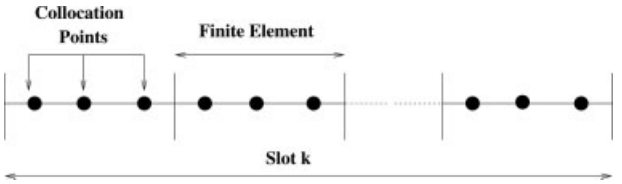


Figure 1. Simultaneous discretization approach for dealing with dynamic optimization problems.

Each slot k is divided into N_{fe} finite elements. Within each finite element f, a set of N_{cp} collocation points c is selected.

d) Timing relations

$$t_k^e = t_k^s + p_k + \theta_k^t, \ \forall k \quad (5a)$$

$$t_k^s = t_{k-1}^e, \ \forall k \neq 1 \quad (5b)$$

$$t_k^e \leq T_c, \ \forall k \quad (5c)$$

2. Dynamic Optimization. To address the optimal control part, the simultaneous approach[11] for dynamic optimization problems was used in which the dynamic model representing the system behavior is discretized using the method of orthogonal collocation on finite elements.[15,16] According to this procedure, a given slot k is divided into a number of finite elements. Within each finite element an adequate number of internal collocation points is selected. Using several finite elements is useful to represent dynamic profiles with nonsmooth variations. Thus, the set of ordinary differential equations comprising the system model, is approximated at each collocation point leading to a set of nonlinear equations that must be satisfied.

A representation of orthogonal collocation is shown in Figure 1. Manipulated variables u in the formulation refer either to feed stream of the reaction system, or cooling water flow rates. On the other hand, systems states x represent concentrations, polymer moments and temperatures, in the reaction system.

a) Dynamic mathematical model discretization

$$x_{fck}^n = x_{o,fk}^n + \theta_k^t h_{fk} \sum_{l=1}^{N_{cp}} \Omega_{lc} \dot{x}_{flk}^n, \ \forall n,f,c,k \quad (6)$$

Also note that in the present formulation the length of all finite elements is the same and computed as

$$h_{fk} = \frac{1}{N_{fe}} \quad (7)$$

b) Continuity constraint between finite elements

$$x_{o,fk}^n = x_{o,f-1,k}^n + \theta_k^t h_{f-1,k} \sum_{l=1}^{N_{cp}} \Omega_{l,N_{cp}} \dot{x}_{f-1,l,k}^n, \ \forall n, f \geq 2, k \quad (8)$$

c) Model behavior at each collocation point

$$\dot{x}_{fck}^n = f^n(x_{fck}^1, \ldots, x_{fck}^n, u_{fck}^1, \ldots u_{fck}^m), \ \forall n,f,c,k \quad (9)$$

The righthand side of Eq. 9 corresponds to the right hand side of the differential equation that describes the

dynamics of state n, solved in collocation point c of finite element f of the discretized transition period of slot k.

d) Time discretization during transition periods

$$t_{fck} = (f-1)\frac{\theta^t_k}{N_{fe}} + \frac{\theta^t_k}{N_{fe}}\gamma_c,\ \forall f,c,k \tag{10}$$

e) Initial and final controlled and manipulated variable values at each slot

$$x^n_{in,k} = \sum_{i=1}^{N_p} x^n_{ss,i} y_{i,k},\ \forall n,k \tag{11}$$

$$\bar{x}^n_k = \sum_{i=1}^{N_p} x^n_{ss,i} y_{i,k+1},\ \forall n, k \neq N_s \tag{12}$$

$$\bar{x}^n_k = \sum_{i=1}^{N_p} x^n_{ss,i} y_{i,1},\ \forall n, k = N_s \tag{13}$$

$$u^m_{in,k} = \sum_{i=1}^{N_p} u^m_{ss,i} y_{i,k},\ \forall m,k \tag{14}$$

$$\bar{u}^m_k = \sum_{i=1}^{N_p} u^m_{ss,i} y_{i,k+1},\ \forall m, k \neq N_s - 1 \tag{15}$$

$$\bar{u}^m_k = \sum_{i=1}^{N_p} u^m_{ss,i} y_{i,1},\ \forall m, k = N_s \tag{16}$$

$$u^m_{1,1,k} = u^m_{in,k},\ \forall m,k \tag{17}$$

$$x^n_{o,1,k} = x^n_{in,k},\ \forall n,k \tag{18}$$

$$x^n_{tol,k} \geq x^n_{Nfe,Nc,k} - \bar{x}^n_k,\ \forall n,k \tag{19}$$

$$-x^n_{tol,k} \leq x^n_{Nfe,Nc,k} - \bar{x}^n_k,\ \forall n,k \tag{20}$$

f) Lower and upper bounds on the decision variables

$$x^n_{min} \leq x^n_{fck} \leq x^n_{max},\quad \forall n,f,c,k \tag{21a}$$

$$u^m_{min} \leq u^m_{fck} \leq u^m_{max},\quad \forall m,f,c,k \tag{21b}$$

g) Smooth transition constraints

$$u^m_{f,c,k} - u^m_{f,c-1,k} \leq u^c_{cont},\ \forall m,k,c \neq 1 \tag{22}$$

$$u^m_{f,c,k} - u^m_{f,c-1,k} \geq -u^c_{cont},\ \forall m,k,f,c \neq 1 \tag{23}$$

$$u^m_{f,1,k} - u^m_{f-1,Nfe,k} \leq u^f_{cont},\ \forall m,k,f \neq 1 \tag{24}$$

$$u^m_{f,1,k} - u^m_{f-1,Nfe,k} \geq -u^f_{cont},\ \forall m,k,f \neq 1 \tag{25}$$

$$u^m_{1,1,k} - u^m_{in,k} \leq u^f_{cont},\ \forall k \tag{26}$$

$$u^m_{1,1,k} - u^m_{in,k} \geq -u^f_{cont},\ \forall k \tag{27}$$

$$\dot{x}^n_{Nfe,Ncp,k} \geq -\dot{x}_{tol,k},\ \forall n,k \tag{28}$$

$$\dot{x}^n_{Nfe,Ncp,k} \leq \dot{x}_{tol,k},\ \forall n,k \tag{29}$$

Equations 22–27 force the change between adjacent collocation points and finite elements to be within a certain acceptable range. Equations 28 and 29 are used to make sure that at the end of the dynamic transition the system is at, or very close to, steady state conditions. The model given by Eqs. 1–29 corresponds to a nonconvex MINLP given the nonlinearities in the objective functions and Eqs. 3b, 6, 8 and 9. Therefore, global optimality can in general not be guaranteed unless a deterministic global optimization algorithm is used, which for the size of these problems is often not computationally feasible.

Outline of the Solution Methodology

The nature of the problem at hand suggests the use of a decomposition technique in which the dynamic optimization problem and the scheduling problem are solved separately. The Lagrangean decomposition technique[1] is the basis of the solution methodology presented in this article. The scheduling and control formulation share the binary variables associated with a production schedule: (1) the variables of transitions durations, and (2) the variable of cycle duration. These variables are substituted by equivalent copies in either the scheduling or the control problem, and a new set of constraints makes the copies equal to the original variables. This new set of constraints is dualized by adding them to the objective function using Lagrange multipliers.[17] This modified SSC formulation is separable into a scheduling subproblem and a control problem, where the sum of the objective functions of the two subproblems (see later Eqs. 32 and 33) represents an upper bound to the objective function of the SSC original problem. Van den Heever et al.[2] propose a heuristic in which the binary variables in the original formulation are fixed so that a lower bound is obtained by solving the resulting NLP. In this article the binary variables obtained in the scheduling subproblem are fixed in the original SSC formulation to obtain an NLP that yields a lower bound in each iteration of the heuristic decomposition algorithm. Once an upper bound and a lower bound are obtained one iteration is completed, after which the Lagrange multipliers are updated. This heuristic algorithm stops once the upper and lower bound converge within a defined tolerance or once the maximum number of iterations is exceeded. Since the algorithm is heuristic, the maximum number of iterations can be determined using different criteria. Generally this number will be set so that it stops when the algorithm does not exhibit any significant progress, but it can also be determined by the point in which bounds begin to degrade or subproblems become infeasible.

A step by step solution methodology is presented below:

1. Solve relaxed scheduling subproblem (Eq. 32) and optimal control subproblem (Eq. 33) with multipliers set to zero.
2. Initialize the Lagrange multipliers using subgradient method (see step 7).
3. Solve scheduling subproblem (Eq. 32) and optimal control subproblem (Eq. 33).
4. Obtain an upper bound as scheduling subproblem (Eq. 32) solution, plus optimal control subproblem (Eq. 33) solution.
5. Fix the values of the binary variables using the solution to the scheduling subproblem (Eq. 32). Solve the original problem (with fixed binaries) to obtain lower bound.
6. If |upper bound – lower bound| $\leq$ tolerance; or maximum number of iterations has been reached, then algorithm stops, else go to next step.
7. The Lagrange multipliers are updated through the subgradient method.

$$u^{k+1} = u^k + t^k(y^k - x^k)$$

and

$$t^{k+1} = \frac{\alpha_k(LD(u^k) - P^*)}{\|y^k - x^k\|^2}$$

Note: All terms of this equation are detailed in the next section.

8. Proceed to next iteration ($k = k + 1$), and go to step 3.

One drawback of this method, is that there is no formal mathematical convergence criteria, and since many integer and/or nonlinear problems can result in significant gaps between upper and lower bounding, the problem will most likely stop at the maximum number of iterations.

In MINLP problems a duality gap may exist between the optimal solution of the decomposed problem and the optimal solution to the original problem.[18] If this is the case, then the two bounds will not converge and the algorithm will reach the maximum number of iterations. If the maximum number of iterations is exceeded, then the best lower bound is reported as the solution. A formal mathematical description of the decomposition technique follows.

Lagrangean decomposition

Guignard and Kim[1] present a Lagrangean decomposition technique in which certain variables are duplicated and set equal by new constraints. These new constraints are then relaxed through Lagrangean relaxation,[17,19] yielding a decomposable model over two or more subsets of constraints. Consider the following mathematical programming problem

$$(P) \quad \max\{fx | Ax \le b, Cx \le d, x \in X\}$$

which is equivalent to

$$(P') \quad \max\{fx | Ay \le b, Cx \le d, x \in X, y = x, y \in Y\}$$

A Lagrangean relaxation is obtained for P' by dualizing the constraint $y = x$. This procedure yields a decomposable problem, thus, the name "Lagrangean Decomposition"

$$\begin{aligned}&(LDu)\\ &\max\{fx + u(y - x) | Cx \le d, x \in X,\\ &Ay \le b, y \in Y\}\\ &= \max\{(f - u)x | Cx \le d, x \in X\}\\ &+ \max\{uy | Ay \le b, y \in Y\}\end{aligned}$$

If the feasible regions are convex, then LDu is an upper bound for P for any given u.[17] Then if all of feasible regions are convex and all of the variables are continuous, the tightest upper bound of LDu is equal to the optimal solution of P

$$P = \min_u LDu$$

In the presence of integer variables and other nonconvexities a duality gap may exist.[18] Since this is the case of the current formulation, the search for an optimum will be performed using an heuristic approach[2] that generates upper bounds by solving a problem of the type LDu, and lower bounds by using a heuristic technique to produce feasible solutions to the original problem P. The multipliers used to solve the subproblems are updated iteratively using the Fisher formula that has proven to work well in practice[20]

$$u^{k+1} = u^k + t^k(y^k - x^k)$$

and

$$t^{k+1} = \frac{\alpha_k(LD(u^k) - P^*)}{\|y^k - x^k\|^2}$$

where t^k is a scalar step size, and α is a scalar usually set between 0 and 2, and then decreased when LDu fails to improve in a fixed number of iterations. P^* is the best known solution, and it can be initialized by using the relaxed solution to the subproblems. This method for updating the multipliers is known as the subgradient method.

Lagrangean decomposition for integer programming

Michelon and Maculan[21] present the extension of Lagrangean decomposition for integer nonlinear programming with linear constraints. Using again the example of problem (P), but in the context of integer programming, we have the following:

$$(P) \quad \max\{fx | Ax \le b, Cx \le d, x \in X\}$$

where X is a set for which the integrality constraints are defined, e.g., $X = \{0,1\}$. The feasible domain of (P) remains unchanged[21] if we eliminate constraint $Ax \le b$, and instead add the constrains

$$y = x, \quad Ay \le b, \quad \text{and} \quad y \in CO(X)$$

where $CO(X)$ represents the Convex Hull of set X. A Lagrangean relaxation is obtained for (P) by dualizing the constraint $y = x$. The same procedure described under the Lagrangean decomposition section of this article can be followed afterward.

The important fact to take notice of is that the copy (y) of the original binary variable (x) is continuous, since the domain of the new variable is the convex hull of X, denoted as $CO(X)$. As mentioned previously, the set $X = \{0,1\}$, and its convex hull includes all real numbers between 0 and 1. This fact is of the utmost importance for the development of the decomposition in this article, since it allows the treatment of binary variables in the scheduling subproblem, while their continuous copies are used in the nonlinear control subproblem.

Scheduling and Control MIDO Reformulation

The problem reformulation consists of four steps.

1. Duplicate key variables

$$z_{ik} = y_{ik}, \quad \forall i,k, y \in B, \quad z \in CO(B) \tag{30a}$$

$$\phi_k^t = \theta_k^t, \quad \forall k \tag{30b}$$

$$Dc = Tc \tag{30c}$$

where

B	Set of binary values $\{0, 1\}$
$CO(B)$	Convex Hull of set B

Equation 30a to c create copies of the sequencing variable, the transition duration variable and the cycle duration variable.

It is important to notice that while y is binary variable ($y \in B$), z can take any value between 0 and 1 ($z \in CO(B)$).[21]

2. Assign one copy of each variable to the scheduling constraints and the other to the dynamic optimization constraints; add necessary extra constraints.

- $\phi_k t$ substitutes θ_k^t in Eqs. 6, and 8, 10.
- The following two equations are duplicates of 2a and 2b

$$\sum_{k=1}^{N_s} z_{ik} = 1, \ \forall i \tag{31a}$$

$$\sum_{i=1}^{N_p} z_{ik} = 1, \ \forall k \tag{31b}$$

- Dc substitutes Tc in the transition terms of the objective function.

3. Equations 30a,b, and c are eliminated and added to the objective function by means of a Lagrangian relaxation.[17] The objective function takes the following form

$$\begin{aligned}\max \Bigg\{ & \sum_{i=1}^{N_p} \frac{C_i^p W_i}{T_c} - \sum_{i=1}^{N_p} \frac{C_i^s (G_i - W_i/Tc)\theta_i}{2} \\ & - \left[\sum_{k=1}^{N_s} \sum_{f=1}^{N_f e} h_{fk} \theta_k^t Q_{max}^m \sum_{c=1}^{N_{cp}} u_{fck}^m \gamma_c \right] \frac{C^r}{D_c} \\ & - \left[\sum_{k=1}^{N_s} \sum_{f=1}^{N_f e} h_{fk} \theta_k^t Q_{max}^I \sum_{c=1}^{N_{cp}} u_{fck}^I \gamma_c \right] \frac{C^I}{D_c} \\ & + \sum_{k=1}^{N_s} \sum_{i=1}^{N_p} \left[\mu_y (z_{ik} - y_{ik}) \right] + \sum_{k=1}^{N_s} \left[\mu_\theta (\phi_k^t - \theta_k^t) \right] \\ & + \mu_{Tc} (Dc - Tc) \Bigg\}\end{aligned}$$

μ_y, μ_θ, μ_{Tc} are the lagrangean multipliers. These quantities are updated after every iteration of the heuristic Lagrangean decomposition algorithm.

4. The formulation is decomposed into a scheduling subproblem and a control subproblem.

- Scheduling Subproblem.

$$\begin{aligned}\max \quad & \Bigg\{ \sum_{i=1}^{N_p} \frac{C_i^p W_i}{T_c} - \sum_{i=1}^{N_p} \frac{C_i^s (G_i - W_i/Tc)\theta_i}{2} \\ & + \sum_{k=1}^{N_s} \sum_{i=1}^{N_p} \left[\mu_y (-y_{ik}) \right] + \sum_{k=1}^{N_s} \left[\mu_\theta (-\theta_k^t) \right] + \mu_{Tc} (-Tc) \Bigg\}\end{aligned}$$

s.t. Equations 2a to 5c (32)

- Control Subproblem

$$\begin{aligned}\max \quad & \Bigg\{ - \left[\sum_{k=1}^{N_s} \sum_{f=1}^{N_f e} h_{fck} \theta_k^t Q_{max}^I \sum_{c=1}^{N_{cp}} u_{fck}^I \gamma_c \right] \frac{C^I}{D_c} \\ & - \left[\sum_{k=1}^{N_s} \sum_{f=1}^{N_f e} h_{fk} \theta_k^t Q_{max}^m \sum_{c=1}^{N_{cp}} u_{fck}^m \gamma_c \right] \frac{C^m}{D_c} \\ & + \sum_{k=1}^{N_s} \sum_{i=1}^{N_p} \left[\mu_y (z_{ik}) \right] + \sum_{k=1}^{N_s} \left[\mu_\theta (\phi_k^t) \right] + \mu_{Tc} (Dc) \Bigg\}\end{aligned} \tag{33}$$

Table 1. Design and Operation Parameters for the MMA Polymerization

$F = 1.0$ m^3/h	$M_m = 100.12$ kg/kgmol
$F_I = 0.0032$ m^3/h	$f^* = 0.58$
$F_{cw} = 0.1588$ m^3/h	$R = 8.314$ kJ/(kgmol · K)
$C_{min} = 6.4678$ kgmol/m^3	$-\Delta H = 57800$ kJ/kgmol
$C_{Iin} = 8.0$ kgmol/m^3	$E_p = 1.8283 \times 10^4$ kJ/kgmol
$T_{in} = 350$ K	$E_I = 1.2877 \times 10^5$ kJ/kgmol
$T_{w0} = 293.2$ K	$E_{fm} = 7.4478 \times 10^4$ kJ/kgmol
$U = 720$ kJ/(h · K · m^2)	$E_{tc} = 2.9442 \times 10^3$ kJ/kgmol
$A = 2$ m^2	$E_{td} = 2.9442 \times 10^3$ kJ/kgmol
$V = 0.1$ m^3	$A_p = 1.77 \times 10^9$ m^3/(kgmol · h)
$V_0 = 0.02$ m^3	$A_I = 3.792 \times 10^{18}$ 1/h
$\rho = 866$ kg/m^3	$A_{fm} = 1.0067 \times 10^{15}$ m^3/(kgmol · h)
$\rho_w = 1000$ kg/m^3	$A_{tc} = 3.8223 \times 10^{10}$ m^3/(kgmol · h)
$C_p = 2$ kJ/(kg · K)	$A_{td} = 3.1457 \times 10^{11}$ m^3/(kgmol · h)
$C_{pw} = 4.2$ kJ/(kg · K)	

s.t. Equations 10 to 29, 31a and 31b with variable y_{ik} substituted for z_{ik} and variable θ_k^t substituted for ϕ_k^t.

Case Studies

In the following section three polymerization reaction examples are used to illustrate the usefulness of the decomposition technique. The solution obtained using the full space problem (direct solution) and the scheduling subproblem are MINLPs solved using DICOPT. The control subproblem of the decomposed formulation and the problem used to generate heuristic lower bounds are NLPs solved using CONOPT3. All models are written in GAMS and solved using a 2.0 GHz machine.

MMA Polymerization

Process description. The MMA polymerization system used in this paper is that described by Silva-Beard et al.[22] The polymerization reactions take place in a CSTR. Table 1 shows the design and operating parameters of the reactor. The mathematical model that describes the bulk free-radical MMA polymerization using AIBN as the initiator is described in the following

$$\frac{dC_m}{dt} = -(k_p + k_{fm}) C_m P_o + \frac{F(C_{min} - C_{in})}{V} \tag{34}$$

$$\frac{dC_I}{dt} = -k_I C_I + \frac{F_I C_{Iin} - F C_I}{V} \tag{35}$$

$$\frac{dT}{dt} = \frac{(-\Delta H) k_p C_m}{\rho C_p} P_0 - \frac{UA}{\rho C_p V} (T - T_j) + \frac{F(T_{in} - T)}{V} \tag{36}$$

$$\frac{dD_0}{dt} = (0.5 K_{tc} + k_{td}) P_0^2 + k_{fm} C_m P_0 - \frac{F D_0}{V} \tag{37}$$

$$\frac{dD_1}{dt} = M_m (k_p + k_{fm}) C_m P_0 - \frac{F D_1}{V} \tag{38}$$

$$\frac{dT_j}{dt} = \frac{F_{cw}(T_{w0} - T_j)}{V_0} + \frac{UA}{\rho C_{pw} V_0} (T - T_j); \tag{39}$$

Table 2. Steady States and Grade Information of the MMA Polymerization Reactor

	State A	State B	State C	State D	State E
C_m (kmol/m^3)	5.5542	5.9653	6.0842	6.2341	6.3245
$T_{reactor}$ (K)	362	351	348	344	342
$X\%$	14	8	6	4	2
MWD (kg/kmol)	15000	25000	30000	39000	48000
F_I (m^3/hr)	2.5×10^{-3}	3.2×10^{-3}	2.9×10^{-3}	2×10^{-3}	1.1×10^{-3}
Demand (kg/hr)	0.8	0.7	1	0.8	0.6
Inv. Cost (\$/hr-kg)	0.10	0.12	0.12	0.15	0.15
Mono. Cost (\$/lt$_{feed}$)	10	10	10	10	10
Init. Cost (\$/lt$_{feed}$)	500	500	500	500	500

where

$$P_0 = \sqrt{\frac{2f^* C_I k_I}{k_{td} + k_{tc}}}$$

$$k_r = A_r e^{-E/RT}, \quad r = p, fm, I, td, tc$$

Five different MMA grades are produced, where each grade corresponds to a different steady state. Table 2 shows the values of the main process variables that correspond to each steady state. Solutions using the full space and the decomposition heuristic are easily computed with little computational effort. This example is included so the decomposition approach is analyzed using a simple example, and the results are validated against the solution found in the full solution space.

Results. In this section the results of the iterative heuristic decomposition technique are presented and compared against those obtained by solving the SSC model in full space (direct solution). Our main interest is analyzing the performance of the decomposition technique in terms of the optimal profit and the computational effort. Table 3 contains values of the main decision variables for the solutions found with and without using the decomposition technique. The values of the sequencing binary variables used to initialize both cases corresponds to the sequence $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$. The initialization values of all other decision variables are also identical between the direct and decomposed formulations.

Figure 2 shows the evolution of the upper and lower bounds during the Lagrangean iterative procedure. A drastic improvement in upper bounding from the relaxed solution is observed. In fact a very tight upper bound is obtained in the iteration, and even slightly improved in subsequent iterations. The heuristic lower bound obtained by fixing the binary variables of the original problem to their corresponding values for the scheduling subproblem in the decomposed formulation reaches its maximum value after two iterations. At this point the gap between bounds is approximately 1.5%. In fact, the algorithm could be said to converge in two iterations with a tolerance of 2% for the gap between bounds. For this illustrative example the algorithm is continued after two iterations to show how the improvement stalls after that point.

Table 3. Direct and Lagrangean Solutions for MMA Example

Algorithm	Obj. [\$/hr]	Opt. Sequence	Cycle [hr]	Trans. [hr]	CPU [s]
Direct	81.6	$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$	149.4	14.9	128
Decomposed	83.6	$B \rightarrow A \rightarrow C \rightarrow E \rightarrow D$	139.9	14.0	107

It is important to say that the duplicated variables that allow the decomposition, namely, cycle time, transition time, and sequencing variables, do not converge to the same values during the algorithm, even though the optimal value of the decomposition (upper bound) converges to the optimal of the original problem (lower bound). This has been reported in previous works,[14,23] and it is due to the effect of using the subgradient method for updating the Lagrange multipliers. This is not a serious problem for the heuristic method used in this work, since the optimal value reported is always determined from the best lower bound. This bound is obtained by fixing binary variables, and keeping all of the original problem constraints. Therefore, the solution obtained represents a feasible solution to the original, nondecomposed problem.

Table 3 shows computational times and optimal solutions for the decomposed and direct approaches (including all six iterations). The CPU solution time for the decomposed formulation is 21 s lower than for the direct formulation, and the optimum found is better. The solution time required using both techniques is relatively low (128 and 106 CPUs) in this illustrative example, but still a 16% reduction in the solution time is achieved. Table 4 shows the size of the direct formulation and of each of the subproblems involved

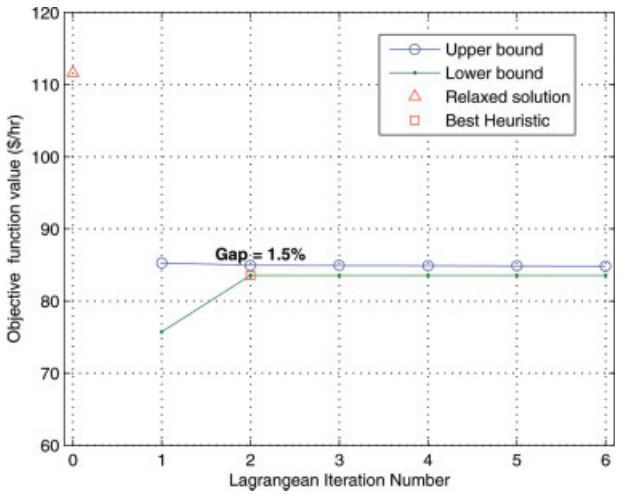


Figure 2. Upper and lower bounds evolution during Lagrangean Heuristic for MMA example.

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

Table 4. Problem Size for Direct and Decomposed Solution for MMA Example

Subproblem	Cont. variables	Discrete Variables	% CPU Time
MMA Direct	4927	25	
MMA Scheduling Subproblem	57	25	0.8
MMA Control Subproblem	4902	none	42
MMA Heuristic Subproblem	4927	none	57.2

Percent of CPU time represents the time spent in each subproblem of the decomposed solution.

in the decomposed formulation. This Table includes the SSC formulation with fixed binary variables, used for obtaining the heuristic lower bound. In this table the increase in number of variables used for the decomposed formulation is evident. However, in the decomposed formulation binary variables are only present in the scheduling subproblem, since the sequencing variables for the control problem are continuous as shown in eq. 30a. Therefore, in the decomposition technique the complicating integer variables are confined to a much smaller problem. From Table 4 one can see that the largest portion of solution time for the decomposition algorithm is spent solving the NLP that provides the lower bounds.

Comparison of optimal solutions. The decomposition technique finds a better local optimum than the one found using the full solution space. Notice how the direct formulation optimum corresponds to the initial sequence provided to the solver. On the other hand, the decomposition results show a very different sequence. The iterative nature of the heuristic decomposition technique, where Lagrange multipliers are updated in every iteration, allows for a different local search every time. The profit of 83.6 $/h is still a local optimum for one set of values of the Lagrangian multipliers.

Reference 4 includes an extensive analysis of the behavior of different decision variables for the isothermal version of the MMA model presented in this work. The findings in ref. 4 can be applied to the comparison between the optimal profit for the direct approach and the decomposed approach. Raw material, inventory costs and product prices are the same as those used in ref. 4. Tables 5 and 6 show the values for the main decision variables for the direct and decomposed solution. Both solution share one important characteristic. Even though there is no upper bound on sales (all polymer produced is sold), all grades are produced exactly to meet demand; not even grade E, which is the most profitable, is produced more than it is demanded. This means that every

Table 5. Simultaneous Scheduling and Control Results in Full Space, for Grade Transition in a MMA Polymerization CSTR

Product	Process T [h]	Production [kg]	Trans T [h]	T start[h]	T end [h]
A	29.88	119.51	2.83	0	32.70
B	20.91	104.57	1.60	62.19	55.22
C	37.35	149.39	1.99	101.24	94.56
D	23.90	119.51	1.69	127.57	120.15
E	22.41	89.63	6.83	127.57	149.39

The objective function value is $ 81.6/h and 149 h of total cycle time.

Table 6. Simultaneous Scheduling and Control Results using Decomposition Heuristic, for Grade Transition in a MMA Polymerization CSTR

Product	Process T [h]	Production [kg]	Trans T [h]	T start [h]	T end [h]
B	19.58	97.89	3.22	0	22.80
A	27.97	111.88	3.18	22.80	53.95
C	34.96	139.85	2.76	53.95	91.67
E	20.98	83.91	1.52	91.67	114.17
D	22.38	111.88	3.30	114.17	139.85

The objective function value is $ 83.6/h and 140 h of total cycle time.

extra hour that the cycle is extended the inventory costs incurred are greater than the profit generated by the extra production. In this scenario the cycle productivity, and the minimization of the time spent in grade transitions becomes very important.

Figures 5 and 6 show the dynamic transitions for the direct and decomposed solutions. The heuristic decomposition algorithm results in an optimal cycle characterized by faster transitions and a shorter, more productive cycle. Figures 5 and 6 show that the longest transition for the direct solution takes almost seven hours, while the longest transition for the decomposed approach takes less than four. For the decomposed solution the optimizer chooses transitions between non sequential grades (A to C, C to E, etc.). This avoids the need for a long transition at the end of the cycle. On the other hand the direct solution chooses transitions between sequential grades (A to B, B to C, etc.), but is forced to make a very long transition (E to A) at the end of the cycle. The optimal trajectories for the decomposed approach show that for a couple of grade transitions the manipulated variable, namely, the initiator flow rate, is completely shut down for a while. This helps in making the transitions cheaper since less initiator is wasted in off-spec material.

HIPS Polymerization Reactor

Process description. A second example is included with the goal of exploring the performance of the proposed

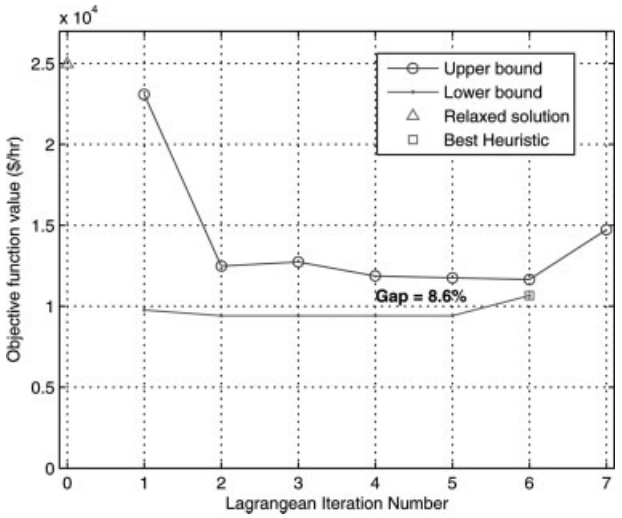


Figure 3. Upper and lower bounds evolution during Lagrangean heuristic for HIPS example.

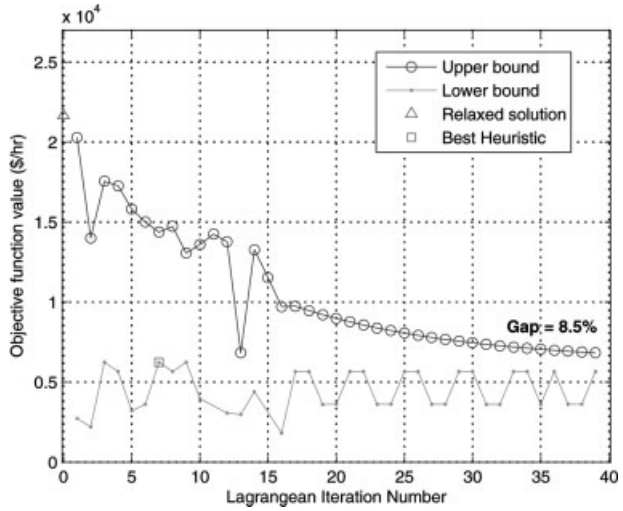


Figure 4. Upper and lower bounds evolution during Lagrangean heuristic for HIPS train example.

approach in presence of a stronger nonlinear behavior and a larger problem. For that purpose the HIPS polymerization system[24] is employed. The polymerization is carried out in a CSTR. Table 7 shows the design and operating parameters of the reactor. The dynamic model of a CSTR where the non-isothermal high impactpolymerization of styrene takes place is given as follows:

- Initiator concentration

$$\frac{dC_i}{dt} = \frac{Q_i C_i^f - QC_i}{V} - K_d C_i \tag{40}$$

- Monomer concentration

$$\frac{dC_m}{dt} = \frac{Q}{V}(C_m^f - C_m) - K_p C_m(\mu_r^o + \mu_b^o) \tag{41}$$

- Butadiene concentration

$$\frac{dC_b}{dt} = \frac{Q}{V}(C_b^f - C_b) - C_b(K_{i2}C_r + K_{fs}\mu_r^o + K_{fb}\mu_b^o) \tag{42}$$

- Radicals concentration

$$\frac{dC_r}{dt} = -\frac{Q}{V}C_r + 2f^* K_d C_i - C_r(K_{i1}C_m + K_{i2}C_b) \tag{43}$$

- Branched radicals concentration

$$\frac{dC_{br}}{dt} = -\frac{Q}{V}C_{br} + C_b(K_{i2}C_r + K_{fb}(\mu_r^o + \mu_b^o)) - C_{br}(K_{i3}C_m + K_t(\mu_r^o + \mu_b^o + C_{br})) \tag{44}$$

- Reactor temperature

$$\frac{dT}{dt} = \frac{Q}{V}(T^f - T) + \frac{\Delta H_r K_p C_m(\mu_r^o + \mu_b^o)}{\rho_s C_{ps}} - \frac{UA(T - T_j)}{\rho_s C_{ps} V} \tag{45}$$

- Cooling jacket temperature

$$\frac{dT_j}{dt} = \frac{Q_w}{V_c}(T_j^f - T_j) + \frac{UA(T - T_j)}{\rho_w C_{pw} V_c} \tag{46}$$

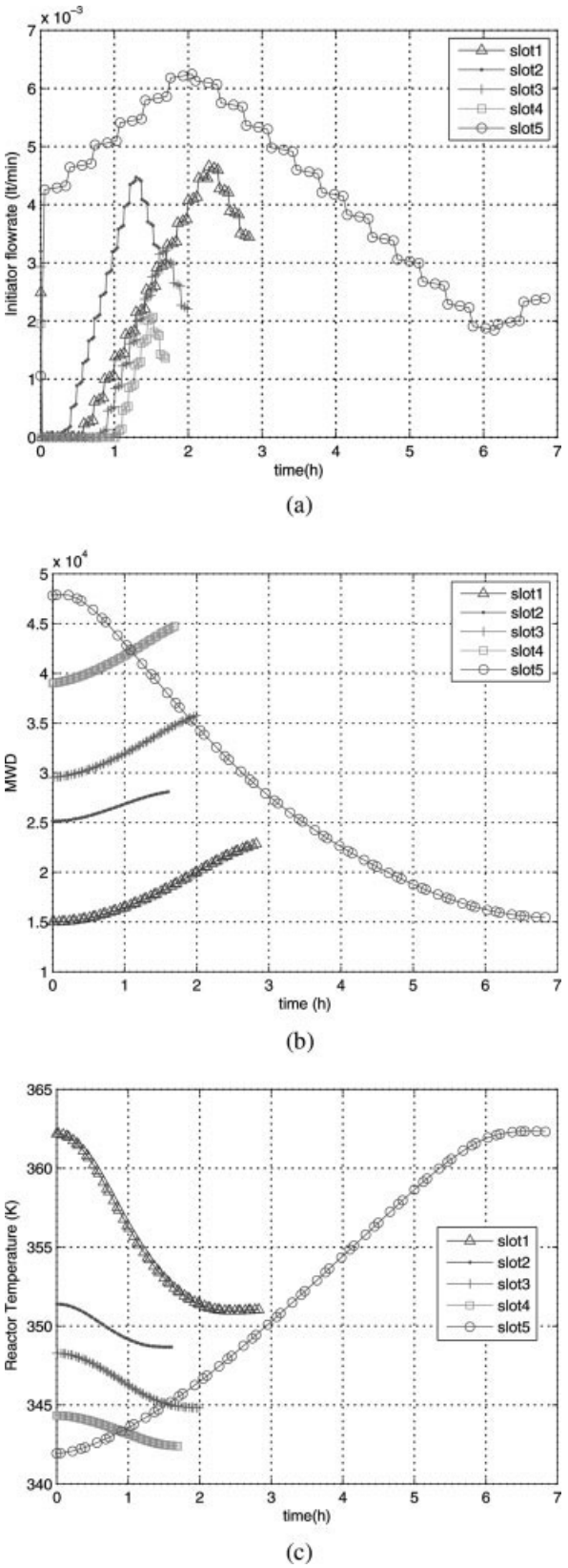
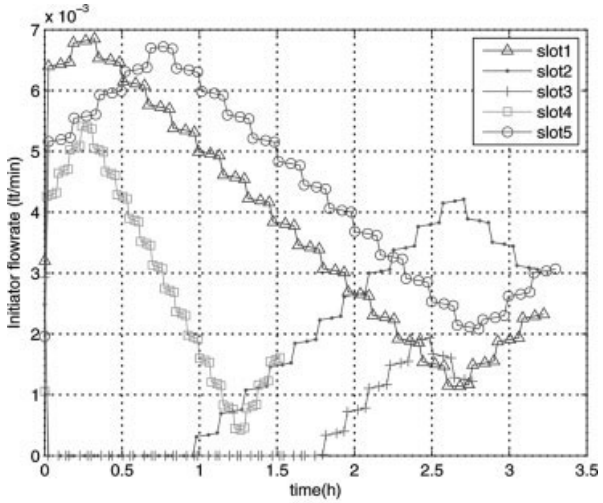
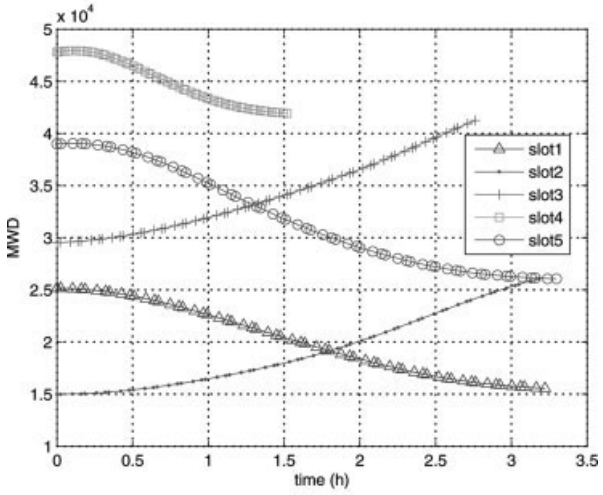


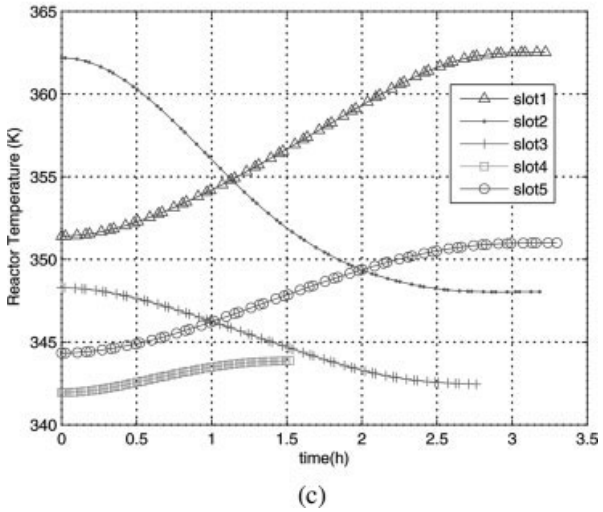
Figure 5. Dynamic transitions for the MMA example obtained by direct solution: (a) Manipulated variable, (b) MWD, and (c) reactor temperature.



(a)



(b)



(c)

Figure 6. Dynamic transitions for the MMA example obtained by Lagrange heuristic: (a) Manipulated variable, (b) MWD, and (c) reactor temperature.

- Zeroth moment live polymer

$$\frac{d\lambda_p^o}{dt} = -\frac{Q}{V}\lambda_p^o + \frac{K_t}{2}(\mu_r^o)^2 + (K_{fs}C_m + K_{fb}C_b)\mu_r^o \tag{47}$$

- First moment live polymer

$$\frac{d\lambda_p^1}{dt} = -\frac{Q}{V}\lambda_p^1 + K_t\mu_r^1\mu_r^o + (K_{fs}C_m + K_{fb}C_b)\mu_r^1 \tag{48}$$

- Zeroth moment dead polymer

$$\frac{d\mu_r^o}{dt} = -\frac{Q}{V}\mu_r^o + 2K_{io}C_m^3 + K_{i1}C_rC_m + C_mK_{fs}(\mu_r^o + \mu_b^o) - (K_pC_m + K_t(\mu_r^o + \mu_b^o + C_{br}) + K_{fs}C_m + K_{fb}C_b)\ \mu_r^o + K_pC_m\mu_r^o \tag{49}$$

- First moment dead polymer

$$\frac{d\mu_r^1}{dt} = -\frac{Q}{V}\mu_r^1 - (K_pC_m + K_t(\mu_r^o + \mu_b^o + C_{br}) + K_{fs}C_m + K_{fb}C_b)\mu_r^1 + K_pC_m(\mu_r^1 + \mu_r^o) \tag{50}$$

- Zeroth moment butadiene

$$\frac{d\mu_b^o}{dt} = -\frac{Q}{V}\mu_b^o + K_{i3}C_{br}C_m - (K_pC_m + K_t(\mu_r^o + \mu_b^o + C_{br}) + K_{fs}C_m + K_{fb}C_b)\mu_b^o + K_pC_m\mu_b^o \tag{51}$$

- Number molecular weight distribution

$$M_n = \frac{\lambda_p^1 + \mu_r^1}{\lambda_p^o + \mu_r^o} \tag{52}$$

Five different HIPS grades are produced, where each grade corresponds to a different steady state. Table 8 shows the values of the main processes variables that correspond to each steady state.

Results

The behavior of the lower and upper bounding of the algorithm is presented in Figure 3, and the optimal solution and key decision variable values are presented in Table 9. From Figure 3, it is clear that there is a significant duality gap between upper and lower bounding. Using this same heuristic algorithm for an oilfield planning problem, Van den Heever et al.[2] found differences of up 13.2% between upper and lower bounds which are similar in magnitude to the 8.6% present in the HIPS case study. From the same Figure, one can observe that the behavior of the upper bound is not always decreasing between sequential iterations. This behavior is normal when the Lagrange multipliers are updated using the subgradient method.[2,14,23] Still there is an overall decrease of the upper bound as iterations increase up to the sixth iteration. Lower bounds do not follow a specific pattern, but this behavior is also expected since they are generated heuristically. After six iterations, the upper bound starts to deteriorate and the lower bounding problem becomes infeasible, so the algorithm is stopped. Two important points observed from the results in Table 9 are that the optimal value found by the decomposition approach is better than the one found in the full solution space, and the computational

Table 7. Nominal Parameter Values for an Industrial Scale HIPS CSTR

Parameter	Value	Units	Parameter	Value	Units
Q	1.1411813934	L/s	T_j^f	294	K
C_i^f	0.9814874815	mol/L	T_j^0	333	K
Q_i	0.0015	L/s	Q_w	1	L/s
C_m^f	8.6310347459	mol/L	C_b^0	1.0547874055	mol/L
V	9450	L	V_c	2000	L
f^*	0.57		ΔH_r	69919.56	J/mol
ρ_s	0.915	kg/L	C_{ps}	1647.265	J/(kg-K)
ρ_w	1	kg/L	C_{pw}	4045.7048	J/(kg-k)
A	19.5	m^2	U	80	J/(s-K-m^2)
R	1.9858	cal/(mol-K)	A_{i0}	1.1×10^5	L^2/(mol^2-s)
E_{i0}	27340	cal/mol	A_d	9.1×10^{13}	l/s
E_d	29508	cal/mol	A_{i1}	1×10^7	L/(mol-s)
E_{i1}	7067	cal/mol	A_{i2}	2×10^6	L/(mol-s)
E_{i2}	7067	cal/mol	A_{i3}	1×10^7	L/(mol-s)
E_{i3}	7067	cal/mol	A_p	1×10^7	L/(mol-s)
E_p	7067	cal/mol	A_t	1.7×10^9	L/(mol-s)
E_t	843	K	A_{fs}	6.6×10^7	L/(mol-s)
E_{fs}	14400	cal/mol	A_{fb}	2.3×10^9	L/(mol-s)
E_{fb}	18000	cal/mol			

time is 38.5% shorter. The key characteristic of the decomposed formulation that leads to shorter solution times for this example is the exclusion of integer variables from the control subproblem. In Table 10, one can see that although the control subproblem is almost as large as the complete direct formulation, its repetitive solution took only 64% of the 1154 CPUs of solution time for the whole decomposition algorithm. This is a much lower solution time per iteration than the 1876 CPUs for the direct formulation.

There are important differences between the MMA and the HIPS cases just presented. As problem size and complexity increase, the duality gap increases, but the benefits in terms of computational time become more significant. The same decomposition and solution methodology was used for the MMA case study and the HIPS case study, so the differences in performance can be attributed to a larger problem size (see Table 10) with stronger nonlinearities and nonconvexities in the HIPS case. Notice, however, that the optimal found by using the decomposition technique is better for both examples.

Comparison of optimal solutions. Table 9 shows that the overall transition time for the optimal solution found by the decomposition algorithm is less than that found by direct solution. Duration of transition times is key in obtaining a better hourly profit for the production cycle, since they represent an unproductive period of the cycle. The cyclic scheduling model has the key feature of continuously depleting products so they are not kept until the end of the cycle. Therefore, the order of production does not influence inventory costs. This being the case, the optimizer will always try to determine the sequence of production, based on the shortest of transitions that in turn renders higher hourly profits. Figures 7 and 9 show the dynamic transitions of both solutions. The shapes of the transition profiles are similar, but the longest transition for the decomposition algorithm takes six hours vs. almost eight hours for the full space solution. The decomposition solution includes transitions between grades that are not adjacent in terms of MWD, like transitions A_2 to A_4 and A_3 to Nm, as opposed to the sequence $Nm \rightarrow A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_4$ chosen by the optimizer using the full solution space. The solution found by means of the decomposition heuristic avoids the need for a very long and expensive transition between the end of one cycle and the beginning of the next. This explains why the longest transition of the decomposed solution is two hours shorter than the longest transition of the direct solution.

The values of some decision variables for both solutions are found in Tables 11 and 12. A common characteristic of both solutions is that the quantity of grade A_3 produced is much greater than what is demanded. The cyclic scheduling formulation used in this work sets only a lower bound for

Table 8. Steady States and Grade Information of the HIPS Polymerization Reactor

	State Nm	State A1	State A2	State A3	State A4
C_m	6.0725	6.6229	6.0842	8.5536	8.600
$T_{reactor}$	389	381	348	330	320
$X\%$	30	23	10	1	0.5
MWD	118000	125000	148000	343000	489000
Q_w [L/s]	1.0	0.5	0.1	0.1	0.5
Demand [kg/hr]	50	60	65	70	60
Price [$/kg]	3.2	4.3	4.5	5.0	5.5
Inv. Cost [$/hr-kg]	0.15	0.20	0.15	0.10	0.25
Mono. Cost [$/$lt_{feed}$]	1	1	1	1	1
Init. Cost [$/$lt_{feed}$]	10	10	10	10	10

Table 9. Direct and Lagrangan Solutions for HIPS Example

Algorithm	Obj. [$/hr]	Opt. Sequence	Cycle [hr]	Trans. [hr]	CPU [s]
Direct	10500.1	$Nm \rightarrow A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_4$	88.4	12.6	1876
Decomposed	10568.2	$A_2 \rightarrow A_4 \rightarrow A_3 \rightarrow N_m \rightarrow A_1$	87.1	12.2	1154

Table 10. Problem Size for Direct and Decomposed Solution for HIPS Example

Subproblem	Cont. Variables	Discrete Variables	% CPU Time
HIPS Direct	13452	25	
HIPS Scheduling Subproblem	62	25	0.07
HIPS Control Subproblem	13422	none	35.62
HIPS Heuristic Subproblem	13452	none	64.31

Percent of CPU time represents the time spent in each subproblem of the decomposed solution.

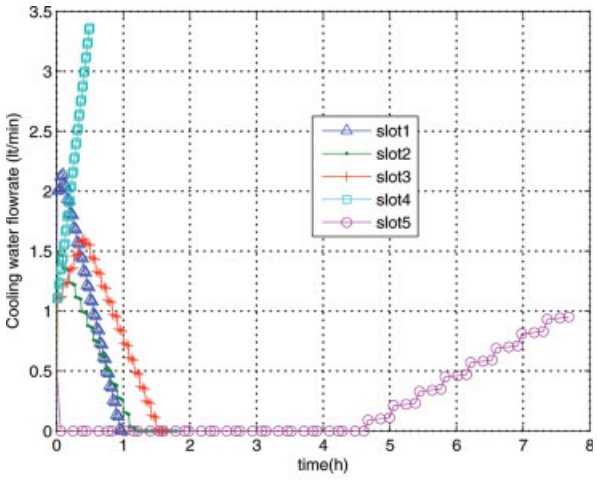
production (demand must be satisfied), but sets no upper bound, since it assumes that all polymer produced is sold. Grade A_3 has the most favorable price to inventory costs proportion (raw material costs are the same for all grades), so it is easy to understand why most of the cycle is devoted to producing this grade, while all other grades are only produced to meet the demands. Also, the cycle resulting from the decomposed solution is shorter and more productive (total transition duration is shorter).

HIPS polymerization reaction train

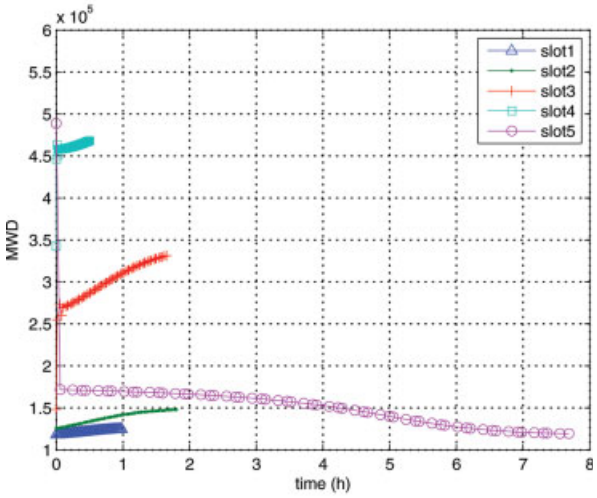
The previous examples were solved using full solution space and using the described decomposition heuristic. The ultimate goal of the decomposition approach is not only to reduce computational effort and time, but to provide a solution for problems that are extremely difficult or even impossible to solve using the full space strategy. The following case study provides an example of a system were we could not obtain a solution using the full solution space.

Process Description. The system used for this case study is the high–impact polystyrene free-radical bulk polymerization system. The mathematical model describing the process is shown in the previous case example. In this example the reaction is not carried out in a single CSTR but in a series of reactors.[25] As displayed in Figure 8, the typical industrial setup to carry out the HIPS polymerization reaction includes a CSTR, followed by a tubular reactor, and a heat exchanger were the reaction is completed. To represent the process without the complexities inherent in the modeling of tubular reactors, a mathematical model consisting of 7 CSTRs is used. The first reactor of the model represents the CSTR in the actual process; the following five CSTRs are used to model the tubular reactor, and the last CSTR in the model represents the heat exchanger. Each CSTR is modeled with a set of differential equations, equivalent to Eqs. 40–46, 49, and 51. Table 13 shows the design parameters for the seven reactors of the model. The rest of the system parameters can be read from Table 7. Four different HIPS grades are produced, where each grade corresponds to a different steady state. Table 14 shows the values of the main processes variables and parameters that correspond to each steady state.

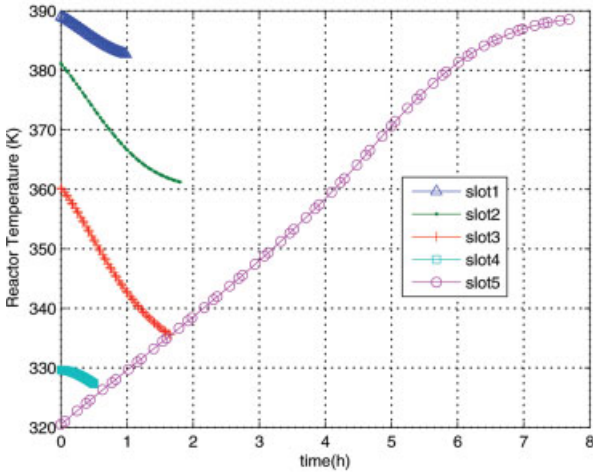
Results and Discussion. Figure 4 shows the evolution of the upper and lower bounding as the Lagrangean iterations proceed. This Figure shows what Guignard[23] describes as a typical behavior of a "good" case when the subgradient



(a)



(b)



(c)

Figure 7. Dynamic transitions for the HIPS example obtained by direct solution: (a) Manipulated variable, (b) MWD, and (c) reactor temperature.

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

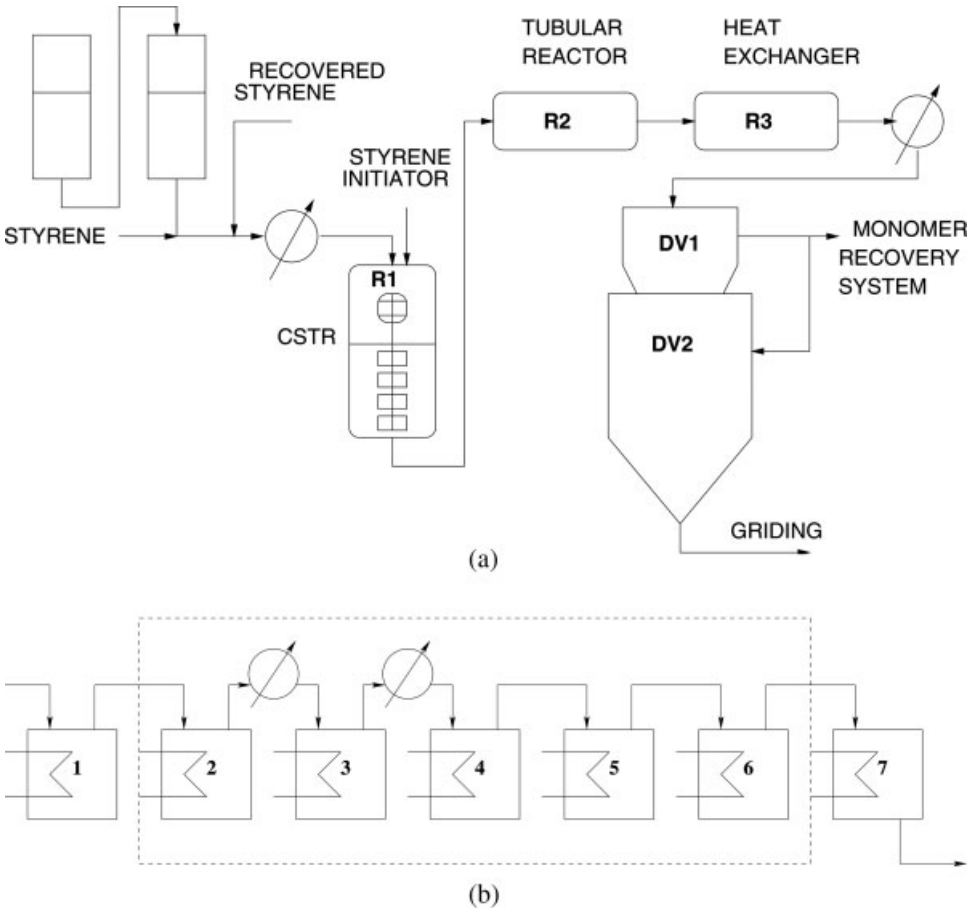


Figure 8. (a) Flow sheet of the HIPS plant, (b) approximation of the HIPS plant by a set of 7 series-connected CSTRs.

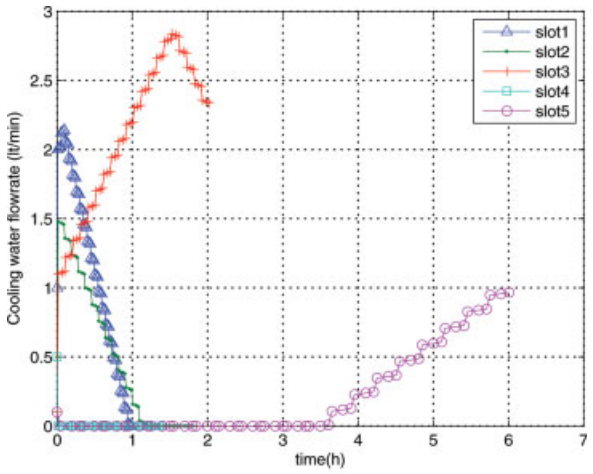
The dashed box stands for the 5 CSTRs employed for approximating the steady-state behavior of the R2 tubular reactor.

method is being used for updating the Lagrange multipliers: a "sawtooth" pattern in the Lagrangean value for the first iterations, followed by a roughly monotonic improvement. The best lower bound, which corresponds to reported optimal solution, is found rather quickly, in less than 10 iterations. However, since the quality of the upper bound is improving considerably when this best lower bound is found, the algorithm is allowed to continue. From iteration 30 onwards, the quality of the upper bound improves very slowly, while the lower bound keeps cycling among a few values. After 39 iterations the algorithm is stopped. At this point there is a 8.5% difference between upper and lower bounding. Once again, the best lower bound is reported as the best solution since all constraints of the original problem are satisfied, and, therefore, it is the best feasible solution found by the algorithm. The optimal solution is reported in Table 15.

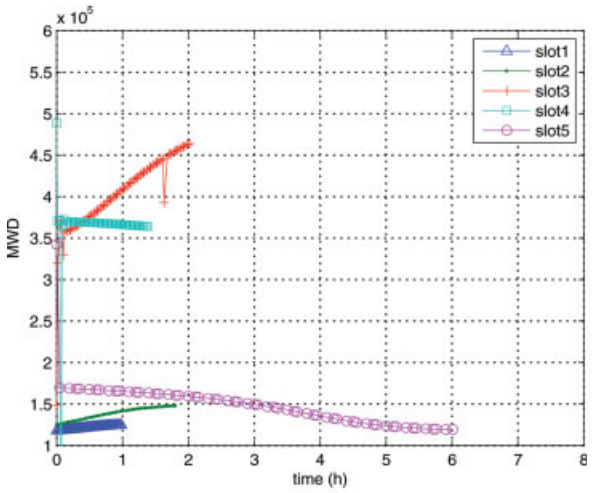
As mentioned in the Solution methodology section, adding more iterations will clearly increase the time required to obtain an heuristic solution. Since there is no formal mathematical convergence criteria in our heuristic methodology, increasing the number of maximum iterations can result in an excessive computational time, which might defeat the whole purpose of the method. Therefore, care must be exercised in the selection of the maximum number of iterations. This is one major drawback of the solution strategy presented in this article.

Values of the most important decision variables for the optimal solution are found in Table 16. One of the key features of the solution is that the production wheel is carried out so that three grades (N, A, B) are produced to satisfy their demands (as required by the formulation), and only grade C, which is the most profitable product, is produced over its demanded quantity (remember that there is no upper bound on production). Transition times are not compared against any other solution, but from previous analysis they are expected to be made as short as possible to achieve a productive manufacturing cycle.

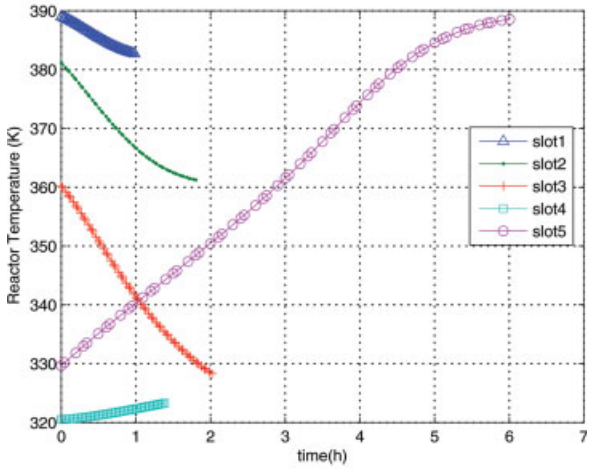
Table 17 shows the size of the problem. In this example the most expensive subproblem is the control subproblem, in contrast to the heuristic subproblem, as in the previous examples. The extra complexity in the dynamic optimization of a train of reactors instead of just one CSTR is the main reason. Notice, once again, that the control subproblem does not include integer variables. The copy of the sequencing variables used in this subproblem is allowed to have continuous values between 0 and 1. The final value for the sequencing



(a)



(b)



(c)

Figure 9. Dynamic transitions for the HIPS example obtained by Lagrange heuristic solution: (a) Manipulated variable, (b) MWD, and (c) reactor temperature.

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

Table 11. Simultaneous Scheduling and Control Results in Full Space, for Grade Transition in a HIPS Polymerization CSTR

Product	Process T [h]	Production [kg]	Trans T [h]	T Start [h]	T End [h]
Nm	2.00	4421.3	0.98	0	2.18
A_1	1.44	5305.6	1.80	2.18	5.42
A_2	1.56	5747.8	1.65	5.42	8.63
A_3	70.18	2.5915e5	0.50	8.63	79.30
A_4	1.44	5305.6	7.69	79.30	88.43

The objective function value is $ 10,500/h and 88.4 h of total cycle time.

variables of the scheduling and the control subproblem for the last iteration of the decomposition algorithm are shown in Table 18. The same Table shows the values for the other duplicated variables for both subproblems. The values of the decision variables shown in this Table are not the optimal values found. The heuristic optimal solution is found in iteration number seven. Nevertheless, it is important to analyze the values of these variables in the last iteration in order to show how the algorithm deals with the duplicity of the variables copied in order to achieve the decomposition. The sequencing variables, y_{ik} for the scheduling subproblem are binary variables, while their copies in the control subproblem z_{ik} are allowed to have continuous values between 0 and 1. This fact is important since it is the key feature that reduces computational effort in the decomposition heuristic when compared against the direct solution. Notice that in the last iteration the values of both sets of variables are identical in some cases and almost the same for others. In the worst cases the continuous variables have values of more than 0.99 or less than 0.01. This was achieved by the term in the objective function that penalizes the differences between copies of sequencing variables. Although the variables in the control subproblem are continuous they are forced by the algorithm to adopt values of exactly or very close to 0 and 1. Moreover, the sequence in the control subproblem is virtually the same as in the scheduling subproblem. The case is not the same for the transition durations and the cyclic time. Both subproblems still do not have the same values for the copied variables. Moreover, there is no mathematical guarantee that all duplicated variables will converge, even if the gap between upper and lower bound is minimized.[14] The objective of the algorithm of finding an optimal solution by heuristic lower bounding, while obtaining reasonable, well

Table 12. Simultaneous Scheduling and Control Results Using Decomposition Heuristic, for Grade Transition in a HIPS Polymerization CSTR

Product	Process T [h]	Production [kg]	Trans T [h]	T Start [h]	T End [h]
A_2	1.53	5659.87	2.02	0	3.55
A_4	1.42	5659.87	1.39	3.55	6.35
A_3	69.34	2.5607e5	6.01	6.35	81.70
Nm	1.18	4353.75	0.98	81.70	83.86
A_1	1.42	5224.59	1.80	83.86	87.08

The objective function value is $ 10,568/h and 87.1 h of total cycle time.

Table 13. Design Parameters for the Seven Reactors of the HIPS Reaction Train

Reactor	Volume [L]	Jacket Volume [L]	Qcw [L/s]	Heat-Transfer Area [m^2]
1	6000	1200	0.1311	11.718
2	900	180	1.0	1.7578
3	1000	200	1.0	1.9531
4	650	130	1.0	1.2695
5	1000	200	1.0	1.9531
6	1000	200	1.0	1.9531
7	5000	1000	1.0	9.5676

Qcw stands for cooling water flow rate.

behaved upper bounds, is considered achieved after a given number of iterations.

The dynamic behaviors of grade transitions in reactors 1, 6 and 7 are shown in Figures 10–13. The profiles of these reactors correspond to the outlet of the actual process equipment: reactor 1 of the model is the actual CSTR, reactor 6 represents the last section of the actual tubular reactor, and reactor 7 of the model represents the process heat exchanger where the reaction is completed. Since the manipulated variable (monomer flow rate) is the same for the seven reactors at any given moment, there is only one dynamic profile per slot. The monomer flow rate during all grade transitions seems to follow the same pattern. The flow rate is decreased during the first part of the transition, and then increased towards its steady-state value. Such initial decrease in flow rate decreases the total cost of the transition since less monomer is wasted as off spec product. From Figures 10–13 one can see that in reactor one the controlled variables undergo small changes in magnitude as they transition from one steady state to the next. This reactor has the lowest conversion values, so the operating temperatures are the lowest. In the initial stage the reaction rate is slower than in the rest of the reactors, and, therefore, the action of the manipulated variable does not have dramatic effects on the controlled variables. In contrast, the temperature and conversion inside reactor six, exhibit very fast responses to changes in the monomer flow rate. Recall that this reactor represents the last segment of the tubular reactor. Since the volume is smaller than in the other two reactors (the residence time is shorter), and the temperatures are high, it is not surprising that the dynamic responses in this reactor are faster. Reactor 7 is similar in size to reactor 1, but its operating temperature is higher, so larger changes in conversion are carried out during grade transition. Duration of dynamic transitions in the HIPS reactor train is similar to dynamic transitions in the previous HIPS example. The longest transition for both examples takes around six hours.

Table 14. Steady States and Grade Information of the HIPS Polymerization Reaction Train

	State N	State A	State B	State C
C_m [mol/L]	3.1344	2.3018	1.4534	0.7519
$T_{reactor}$ [K]	395	440	476	517
$X\%$	64	73	83	91
Q_w [L/s]	1.14	1.48	1.64	2.10
Demand [kg/hr]	350	325	300	250
Price [$/kg]	3.2	4.3	4.5	5.0
Inv. Cost [$/hr-kg]	0.16	0.21	0.22	0.25
Mono. Cost [$/lt$_{feed}$]	1	1	1	1
Init. Cost [$/lt$_{feed}$]	100	100	100	100

State values and conversion correspond the last CSTR of the reaction train.

Table 15. Lagrangean Solution for HIPS Train Example

Algorithm	Obj. [$/h]	Opt. Sequence	Cycle [h]	Trans. [h]	CPU [s]
Decomposed	6244.5	$A \rightarrow N \rightarrow C \rightarrow B$	39.2	12.2	10600

This last example illustrates how the decomposition heuristic described in this work represents a practical alternative for solving SSC problems, especially as they grow in size, nonlinearity and nonconvexity. In previous examples, although the computational effort of the decomposition approach was reduced, the solution in full space was available. However, in this last example it was impossible to find a solution in full space, using the same outer approximation algorithm (DICOPT) that was successfully used in previous examples. When trying to obtain a solution without the decomposition approach, the MINLP problem invariably became infeasible.

In order to show the contribution of our decomposition strategy, a sequential strategy of solving the 7 CSTR system was used.

Comparison vs. sequential solution. In the sequential solution presented in this section, the SSC problem was decoupled into a dynamic optimization problem, containing everything related to the dynamic behavior of the system during transitions; and a scheduling problem into which transition costs and transition times obtained from the dynamic optimization problem were fed as fixed parameters. The first step of the sequential solution involves calculating all possible dynamic transitions involving grades N, A, B, C. These transitions represent purely dynamical optimization problems, initialized with similar values as those used for the same variables in the decomposed solution presented in previous sections. All constraints of the mathematical formulation used correspond to those presented in the dynamic optimization part of the scheduling and control MIDO formulation in this article. The objective function minimizes transition costs, quantified as the amount of raw materials spent during the transition, from

Table 16. Simultaneous Scheduling and Control Results Using Decomposition Heuristic, for Grade Transition in a HIPS Polymerization Train

Product	Process T [h]	Production [kg]	Trans T [h]	T start[h]	T end [h]
A	3.72	12742.20	5.96	0	8.62
N	2.66	13722.37	3.37	8.62	15.72
C	18.40	1.2501e5	1.43	15.72	35.54
B	2.22	11762.03	1.45	35.54	39.21

The objective function value is $ 6,245/h and 39.2 h of total cycle time.

Table 17. Problem Size for Direct and Decomposed Solution for HIPS Train Example

Subproblem	Cont. variables	Discrete Variables	% CPU Time
HIPS Scheduling Subproblem	46	16	0.03
HIPS Control Subproblem	22678	none	67.76
HIPS Heuristic Subproblem	22702	none	32.21

Percent of CPU time represents the time spent in each subproblem of the decomposed solution.

which off-spec products are obtained. Such an objective function has the following form

$$\min\left\{\left[\sum_{f=1}^{N_f e} h_f \theta^t Q^m_{max} \sum_{c=1}^{N_{cp}} u^m_{fc}\gamma_c\right] C^r + \left[\sum_{f=1}^{N_f e} h_f \theta^t Q^l_{max} \sum_{c=1}^{N_{cp}} u^l_{fc}\gamma_c\right] C^l\right\}$$

Symbols in this equation represent the same as throughout the paper, and can be found in the Appendix. However, the subscript k has been eliminated in terms θ, h_{fck}, u^m_{fck}, and u^l_{fck}, since no slots are used in the pure dynamic optimization of grade transitions.

The resulting grade transition costs and durations are found in Tables 19 and 20 as obtained by CONOPT3/GAMS. In a second step these values are fed as fixed parameters into a scheduling formulation. Such formulation is very similar to the one outlined in the Scheduling part of the Scheduling and Control MIDO formulation section of this article, and it can be found in.[26] The resulting scheduling formulation is a MINLP problem, and it was solved using DICOPT/GAMS.

Table 18. Value of the Duplicated Variables in the Last Lagrangean Iteration

Variable	Scheduling Subproblem	Control Subproblem
y_{B1}	1	–
y_{A2}	1	–
y_{N3}	1	–
y_{C4}	1	–
z_{B1}	–	0.994
z_{A2}	–	0.994
z_{N3}	–	1.000
z_{C4}	–	1.000
z_{A1}	–	0.006
z_{B2}	–	0.006
θ^t_1	0.50	–
θ^t_2	0.50	–
θ^t_3	0.50	–
θ^t_4	0.50	–
ϕ^t_1	–	1.43
ϕ^t_2	–	6.60
ϕ^t_3	–	3.31
ϕ^t_4	–	1.43
Tc	16.634	–
Dc	443.486	–

The best upper bound is found in this last iteration, corresponding to an objective value of $ 6823.58/h. Binary variables, and their continuous equivalents in the control subproblem, not shown have a value of 0.

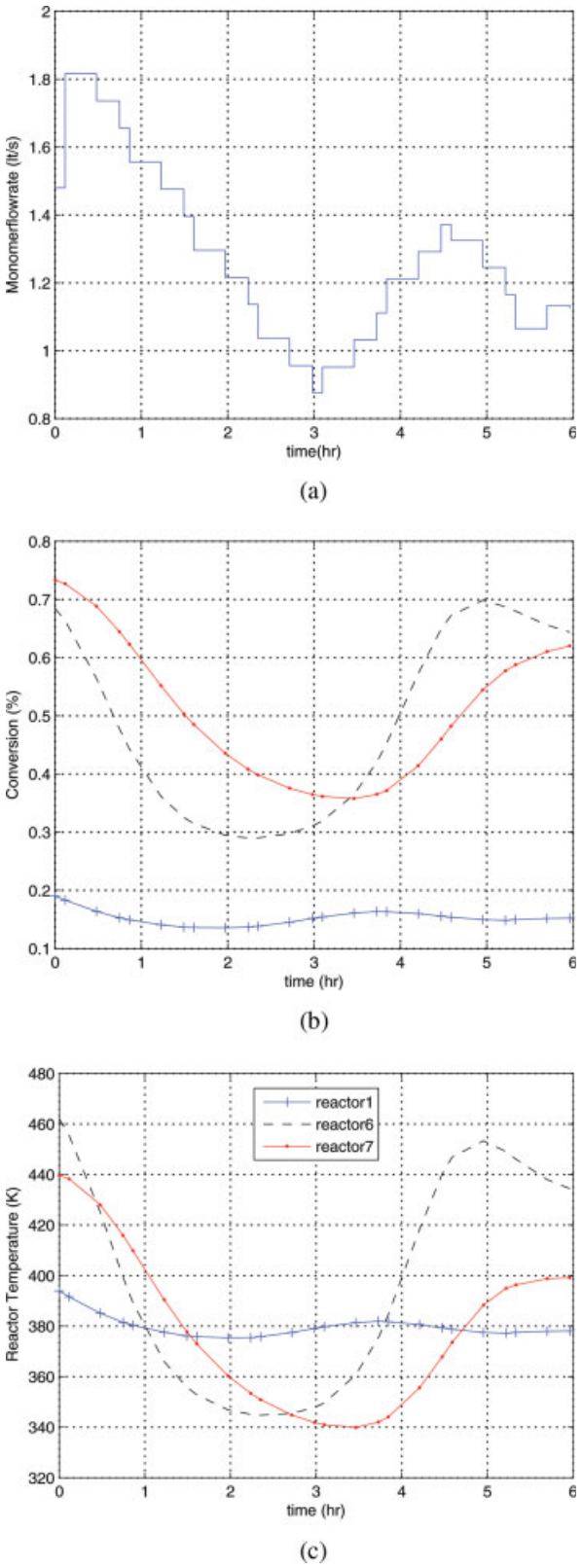
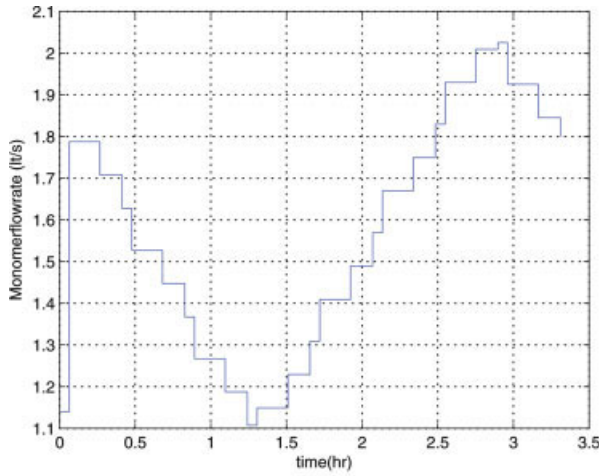
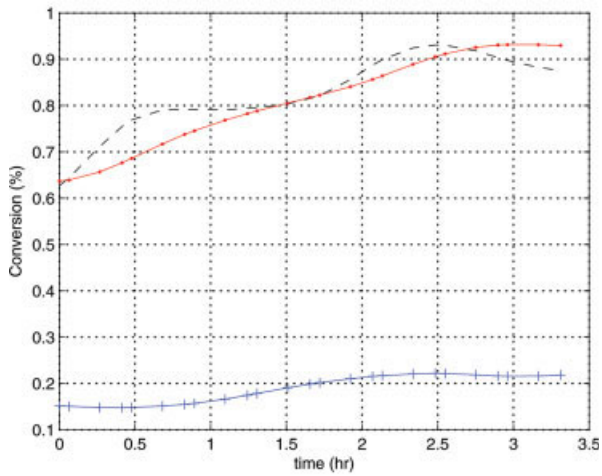


Figure 10. Monomer feed stream (a), conversion (b), and temperature profiles (c) of reactors 1,6 and 7 in slot 1.

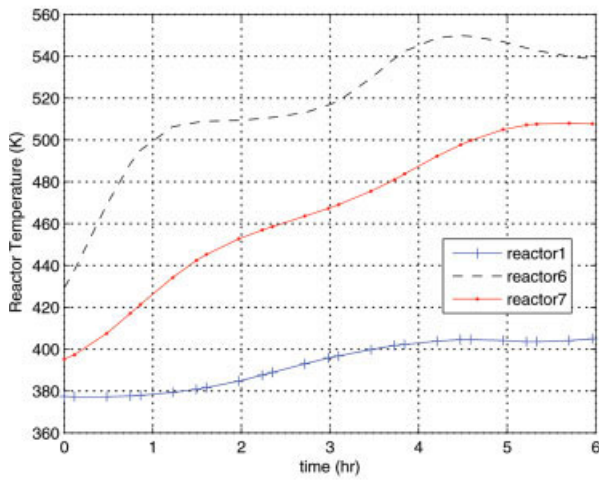
[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]



(a)



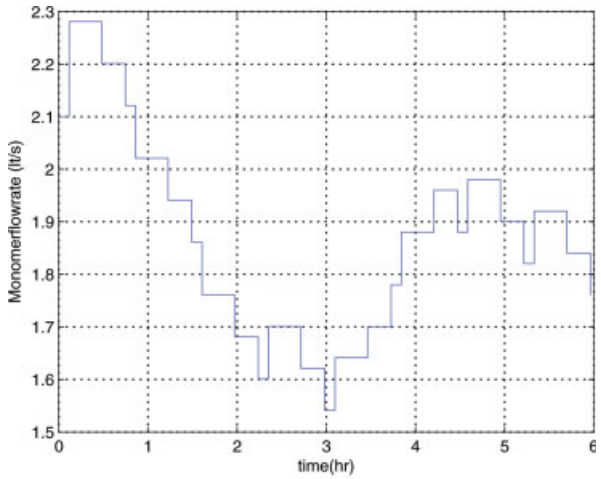
(b)



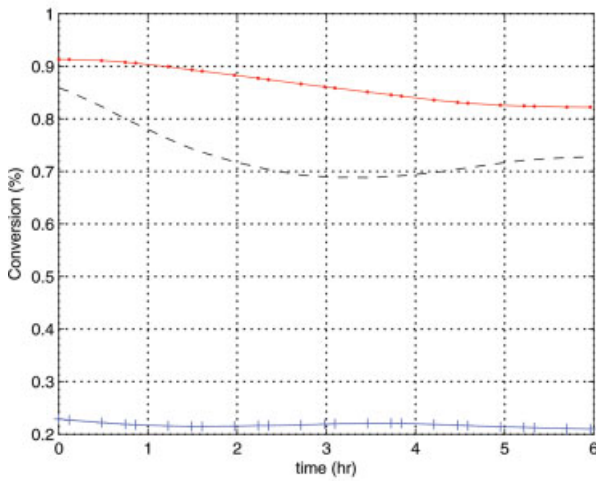
(c)

Figure 11. Monomer feed stream (a), conversion (b), and temperature profiles (c) of reactors 1,6 and 7 in slot 2.

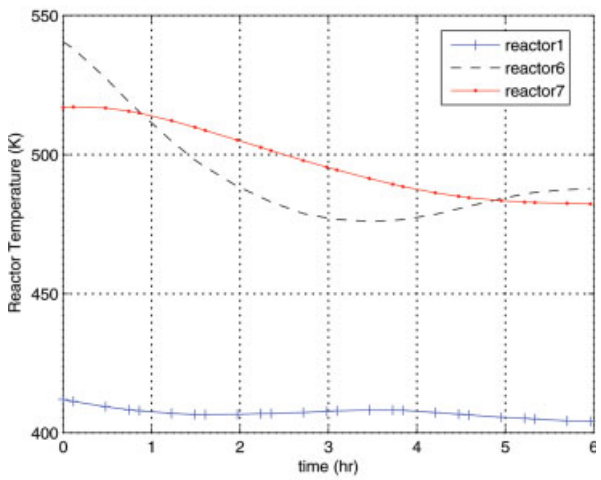
[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]



(a)



(b)



(c)

Figure 12. Monomer feed stream (a), conversion (b), and temperature profiles (c) of reactors 1,6 and 7 in slot 3.

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

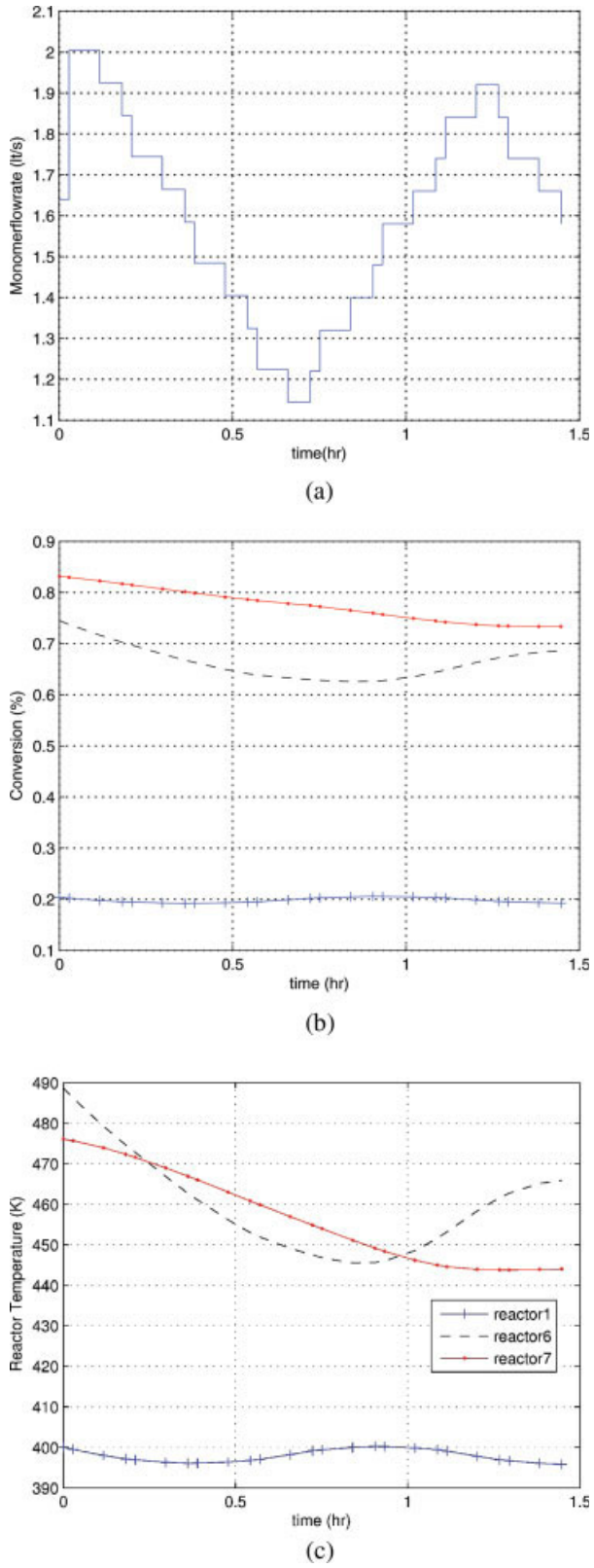


Figure 13. Monomer feed stream (a), conversion (b), and temperature profiles (c) of reactors 1,6 and 7 in slot 4.

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

Table 19. Transition Costs in $ Obtained by Pure Dynamic Optimization of Grade Transitions, During the First Step of the Sequential Approach

Grade	N	A	B	C
N	–	27425	24727	23319
A	34728	–	7785	13146
B	37227	9885	–	8720
C	44937	20981	10650	–

Rows represent the initial grade and columns the final grade for the transitions.

In Table 21 it is evident that even though the sequential solution consumes much less computational effort, the optimal solution found is significantly worse than that obtained using the heuristic decomposition strategy on the SSC problem. Table 22 shows values of some relevant variables at the solution. By comparing Table 16 and Table 22 one can see that most scheduling variables are very similar among both solutions, and that the main differences between the sequential and simultaneous approaches are found in the duration of dynamic transitions. An analysis of this difference between sequential and simultaneous solutions for scheduling and optimal control is found in a previous work.[4] The economic profit found by the heuristic lagrangean decomposition technique is 32% better than that found using the sequential or "intuitive" decomposition technique. Such a difference justifies the extra effort spent both, in the formulation of the problem and in obtaining the solution.

Conclusions

The Lagrangean decomposition methodology as presented by Guignard and Kim[1] was used to reformulate the SSC problem.[4,10] A solution methodology very similar to the one presented by Van den Heever et al.[2] was used. According to this methodology the decomposed formulation is used to generate an upper bound, and a heuristic procedure is used to generate a lower bound. The decomposition approach was successful for solving the SSC problem in the MMA example, and in the HIPS polymerization system, using one CSTR and seven CSTRs. In the first two cases the optimal solution found by the decomposition approach was better than the solution obtained in the full space. In these two examples, the computational effort required by the decomposition heuristic is lower than the computational effort required by the direct solution (solution in full space). The third problem was only solvable using the decomposition heuristic. This work does not prove that the solution to the SSC problem in the HIPS reactor train example cannot be obtained without the decom-

Table 20. Transition Durations in Hours Obtained by Pure Dynamic Optimization of Grade Transitions, During the First Step of the Sequential Approach

Grade	N	A	B	C
N	–	5.25	4.50	3.68
A	6.87	–	1.30	1.91
B	7.29	1.58	–	1.20
C	8.58	3.20	1.48	–

Rows represent the initial grade and columns the final grade for the transitions.

Table 21. Comparison Between Lagrangean Decomposition and Sequential Solutions for HIPS Polymerization Train

Method	Solution Time [CPU s]	Profit [$/h]
Simultaneous	10600	6244.5
Sequential	217	4717.1

Table 22. Sequential Scheduling and Control Results for Grade Transition in a HIPS Polymerization Train

Product	Process T [h]	Production [kg]	Trans T [h]	T Start [h]	T End [h]
B	2.32	12316.76	1.59	0	3.91
A	2.79	13343.16	6.87	3.91	13.56
N	3.90	14369.56	3.68	13.56	21.13
C	18.45	1.2537e5	1.48	21.13	41.06

The objective function value is $ 4717 and 41 h of total cycle time.

position. However, it does show is that if such a direct solution is available, the effort required to obtain it becomes unpractical.

Other decomposition techniques for MIDO problems are available, like Benders decomposition split the problem into a MILP and a NLP problem, and then iterate between the two. A particular example[8] in which the SSC problem is decoupled into a scheduling master problem and an optimal control primal problem, has similar solution times to the MMA example presented in this article, but, since the nonlinear model is not shown in detail, there is no way to make a rigorous comparison between their method and ours. To the best of the authors knowledge there are no further works that solve the SSC problem by using decomposition techniques, and, therefore, it is not possible at this point to make an analysis of the relative performance of the methodology proposed in this work.

Literature Cited

1. Guignard M, Kim S. Lagrangean Decomposition: A model yielding stronger Lagrangean bounds. *Math Progr.* 1987;39:215–228.
2. Van-Den-Heever SA, Grossmann IE, Vasantharajan S, Edwards K. A Lagrangean decomposition heuristic for the design and planning of offshore hydrocarbon field infrastructures with complex economic objectives. *Ind Eng Chem Res.* 2001;40:2857–2875.
3. Mishra BV, Mayer E, Raisch J, Kienle A. Short-term scheduling of batch processes. A comparative study of different, approaches. *Ind Eng Chem Res.* 2005;44:4022–4034.
4. Terrazas-Moreno S, Flores-Tlacuahuac A, Grossmann IE. Simultaneous cyclic Scheduling and optimal control of polymerization reactors. *AICHE J.* 2007;53(9):2301–2315.
5. Mahadevan R, Doyle FJ III, Allcock AC. Control-relevant scheduling of polymer grade transitions. *AICHE J.* 2002;48(8):1754–1764.
6. Feather D, Harrell D, Liberman R, Doyle FJ. Hybrid approach to polymer grade transition control. *AICHE J.* 2004;50(10):2502–2513.
7. Prata A, Oldenburg J, Marquardt W, Nystrom R, Kroll A. Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. accepted for publication, Comput. Chem Eng. 2007, doi:10.1016/j.compchemeng.2007.03.009.
8. Nystrom RH, Franke R, Harjunkoski I, Kroll A. Production campaign planning including grade transition sequencing and dynamic optimizatioin. *Comput Chem Eng.* 2005;29(10):2163–2179.
9. Nystrom RH, Harjunkiski I, Kroll A. Production optimization for continuously operated processes with optimal operation and scheduling of multiple units. *Comput Chem Eng.* 2006;30:392–406.
10. Flores-Tlacuahuac A, Grossmann IE. Simultaneous cyclic scheduling and control of a multiproduct CSTR. *Ind Eng Chem Res.* 2006;45(20): 6698–6712.
11. Biegler LT. Optimization strategies for complex process models. *Adv in Chem Eng.* 1992;18:197–256.
12. deMatta R. A Lagrangean decomposition solution to a single line multiproduct scheduling problem. *Euro J of Oper Res.* 1994;79:25–37.
13. Wu D, Ierapetritou MG. Decomposition approaches for the efficient solution of short-term scheduling problems. *Comput Chem Eng.* 2003; 27:1261–1276.
14. Wu D, Ierapetritou M. Lagrangean decomposition using an improved Nelder-Mead approach for Lagrangean multiplier update. *Comput Chem Eng.* 2006;30(5):778–789.
15. Finlayson B. *Nonlinear Analysis in Chemical Engineering*. New York: McGraw-Hill. 1980.
16. Villadsen J, Michelsen ML. *Solution of Differential Equations Models by Polynomial Approximation*. Prentice-Hall. 1978.
17. Fisher ML. The Lagrangean relaxation method for solving integer programming problems. *Management Sci.* 1981;27(1):1.
18. Guignard M. Lagrangean relaxation: A short course. *Belg J. OR.: Special Issue Francoro.* 1995;35:3.
19. Geoffrion AM. Lagrangean relaxation for integer programming. *Math Program Study 2*. 1974;2:82–114.
20. Fisher ML. An application oriented guide to Lagrangian relaxation. *Interfaces.* 1985;15(2):10.
21. Michelon P, Maculan N. Lagrangean decomposition for integer nonlinear programming with linear constraints. *Math Program.* 1991;52: 203–313.
22. Silva-Beard S, Flores-Tlacuahuac A. Effects of Process design/operation on the steady state operability of a methyl-methacrylate polymerization reactor. *Ind Eng. Chem Res.* 1999;38:4790–4804.
23. Guignard M. Lagrangean relaxation. *Top.* 2003;11(2):151–228.
24. Flores-Tlacuahuac A, Verazaluce-García JC, Saldívar-Guerra E. Steady-state non-linear bifurcation analysis of a high impact polystyrene continuous stirred reactor. *Ind Eng Chem Res.* 2000;39:1972–1979.
25. Flores-Tlacuahuac A, Biegler LT, Saldívar-Guerra E. Dynamic optimization of HIPS open-loop unstable polymerization reactors. *Ind Eng Chem Res.* 2005;44:2659–2674.
26. Pinto JM, Grossmann IE. Optimal cyclic scheduling of multistage continuous multiproduct plants. *Comput Chem Eng.* 1994;18(9):797–816.

Appendix

The indices, decision variables and system parameters used in the SSC MIDO problem formulation are as follows:

Indices

products	$i, p = 1, \ldots N_p$
slots	$k = 1, \ldots N_s$
finite elements	$f = 1, \ldots N_{fe}$
collocation points	$c,l = 1, \ldots N_{cp}$
system states	$n = 1, \ldots N_x$
manipulated variables	$m = 1, \ldots N_u$

Decision variables

y_{ik} = binary variable to denote if product i is assigned to slot k
z_{ik} = copy of y_{ik}
p_k = processing time at slot k
t_k^e = final time at slot k
t_k^s = start time at slot k
t_{fck} = time at finite element f and collocation point c of slot k
G_i = production rate
Tc = cyclic time h
Dc = copy of Tc
x_{fck}^n = n-th system state in finite element f and collocation point c of slot k
$\dot{x}_{flk}^n$ = value of n-th state derivative with respect to time in finite element f and collocation point l of slot k
u_{fck}^m = m-th manipulated variable in finite element f and collocation point c of slot k

W_i = amount produced of each product kg
θ_{ik} = processing time of product i in slot k
θ^t_k = transition time at slot k
ϕ^t_k = copy of θ^t_k
Θ_i = total processing time of product i
$x^n_{o,fk}$ = n-th state value at the beginning of the finite element f of slot k
$\bar{x}^n_k$ = desired value of the n-th state at the end of dynamic transition of slot k
$\bar{u}^m_k$ = desired value of the m-th manipulated variable at the end of dynamic transition of slot k
$x^n_{in,k}$ = n-th state value at the beginning of dynamic transition of slot k
$u^n_{in,k}$ = m-th manipulated variable value at the beginning of dynamic transition of slot k
X_{fck} = conversion in finite element f and collocation point c of slot k
MWD_{fck} = molecular Weight Distribution in finite element f and collocation point c of slot k

Parameters

N_p = number of products
N_s = number of slots
N_{fe} = number of finite elements
N_{cp} = number of collocation points
N_x = number of system states
N_u = number of manipulated variables
C^f_m = monomer feed stream concentration kmol/m^3
D_i = demand rate kg/h
C^p_i = price of products \$/kg
C^s_i = cost of inventory \$/kg-h
C^r = cost of raw material \$/L of feed solution
C^I = cost of initiator \$/L of feed solution
h_{fk} = length of finite element f in slot k
Ω_{cc} = matrix of Radau quadrature weights
$\bar{x}^n_k$ = desired value of the n-th system state at slot k
$\bar{u}^m_k$ = desired value of the m-th manipulated variable at slot k
θ^{max} = upper bound on processing time
$x^n_{ss,i}$ = n-th state value at steady state of product i
$u^m_{ss,i}$ = m-th manipulated variable value of product i
F^o_i = feed stream volumetric flow rate at steady state for grade i m^3/hr
$MW_{monomer}$ = monomer molecular weight kg/kmol
$X_{ss,i}$ = desired conversion degree of grade i
$MWD_{ss,i}$ = desired molecular weight distribution of grade i
x^n_{min}, x^n_{max} = minimum and maximum value of the state x^n
u^m_{min}, u^m_{max} = minimum and maximum value of the manipulated variable u^m
$\dot{x}^n_{tol}$ = maximum absolute value for state derivatives at the end of dynamic transition
x^n_{tol} = maximum absolute deviation from desired final value, allowable for state variable x^n at the end of dynamic transition
u^f_{cont} = maximum absolute change for u^m between finite elements
u^c_{cont} = maximum absolute change for u^m between collocation points
γ_c = roots of the Lagrange orthogonal polynomial
Q^m_{max} = maximum monomer flow rate L/h
Q^I_{max} = maximum initiator flow rate L/h

Manuscript received Apr. 16, 2007, and revision received Sept. 9, 2007.