Simultaneous Scheduling and Control of Multi-Grade Polymerization Reactors

Antonio Flores-Tlacuahuac

Department of Chemical Engineering

Universidad Iberoamericana

México, DF

mailto:antonio.flores@uia.mx

June 24, 2008

Outline

- Introduction
- Problem Definition
- Mixed-Integer Dynamic Optimization Formulation
- Extension to Handle Grade Transitions in Polymerization Reactors
- Decomposition Optimization Approach to Solve Larger Size MINLP Problems
- Conclusions
- Future Work

Introduction

Traditionally Scheduling and Control problems are solved independently.

- Scheduling:

  From a scheduling point of view, the interest lies in determining optimal assignments to equipment, production sequences, production times for each product, and inventory levels that lead to maximum profit or minimum completion time. Commonly, during this task, the dynamic behavior of the underlying process is not taken into account.

- Control:

  Similarly, when computing optimal transition trajectories between different set of products, one of the major objectives lies in determining the transition trajectory featuring minimum transition time. When addressing optimal control problems, it is normally assumed that the production sequence is fixed. Hence, scheduling decisions are normally neglected in optimal control formulations.

In scheduling problems, the transition times between the different product combinations are assumed to be known as fixed values, and hence, the dynamic profile of the chosen manipulated and controlled variables is not taken into account in the optimization formulation. It has been recognized, however, that scheduling and control problems are closely related problems and that, ideally, they should be addressed simultaneously rather than sequentially or solved without taking into account both parts.

Aim of this talk

In this work, we propose a simultaneous approach to address scheduling and control problems for a set of continuous plants. We take advantage of the rich knowledge of scheduling and optimal control formulations, and we merge them so the final result is a formulation able to solve simultaneous scheduling and control problems.

We cast the problem as an optimization problem. In the proposed formulation:

- Integer variables are used to determine the best production sequence
- Continuous variables take into account production times, cycle time, and inventories

Because dynamic profiles of both manipulated and controlled variables are also decision variables, the resulting problem is cast as a mixed-integer dynamic optimization (MIDO) problem.

To solve the MIDO problem, we use a methodology that consists of transforming the MIDO problem into an MINLP that can be solved using standard methods. The strategy for solving the MIDO problem consists of using the so-called simultaneous approach for solving optimal control problems as the way to transform the set of ordinary differential equations modeling the dynamic system behavior into a set of algebraic equations. Because of the highly nonlinear behavior embedded in chemical process models, the resulting MIDO formulation will be an MINLP problem featuring difficult nonlinearities such as multiple steady states and parametric sensitivity.

Problem Definition

Given are:

- A number of products to be manufactured in a single CSTR
- Lower bounds for the product demands
- Steady-state operating conditions for each desired product
- Cost of each product
- Inventory and raw materials costs

The problem consists in:

> *Simultaneous determination of a cyclic schedule (i.e. production wheel) and the control profile such that a given cost function is minimized*

Major decisions involve:

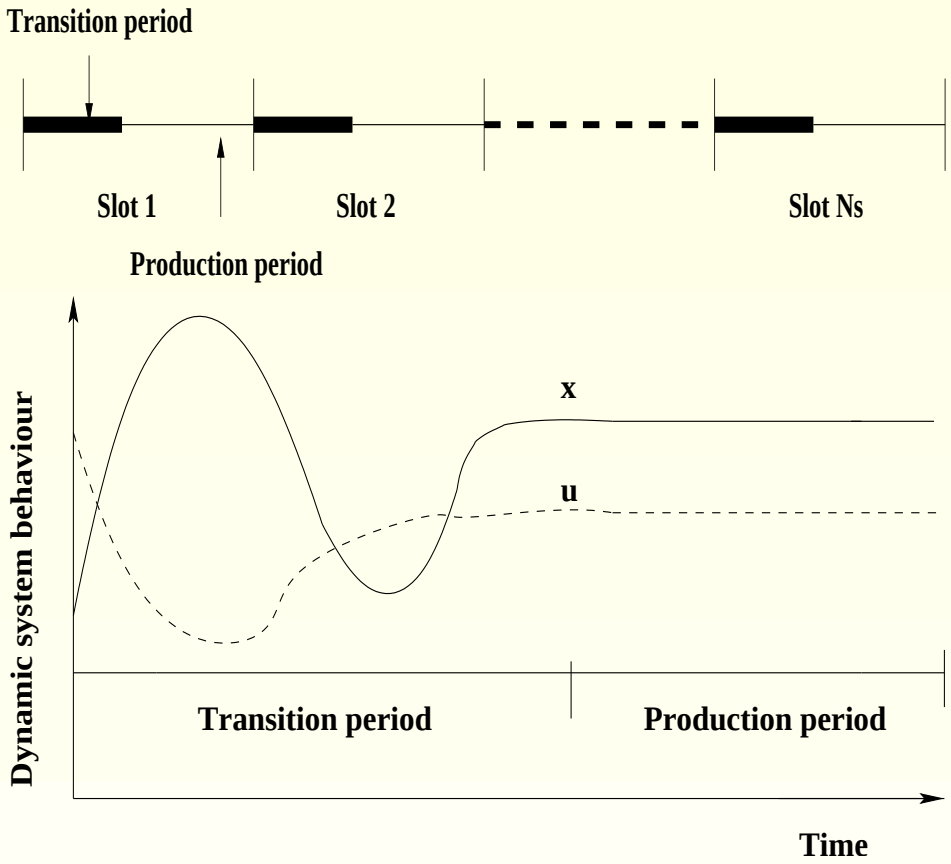
- Selecting the cyclic time and Sequence in which the products will be manufactured
- The transition times, Production rates, Length of processing times
- Amounts manufactured of each product
- Manipulated variables profiles for the transition

Problem assumptions:

- All products are manufactured in a single CSTR
- Products sequence follows a production wheel
- Cyclic time is divided into a series of slots

Two operations occur inside each slot:

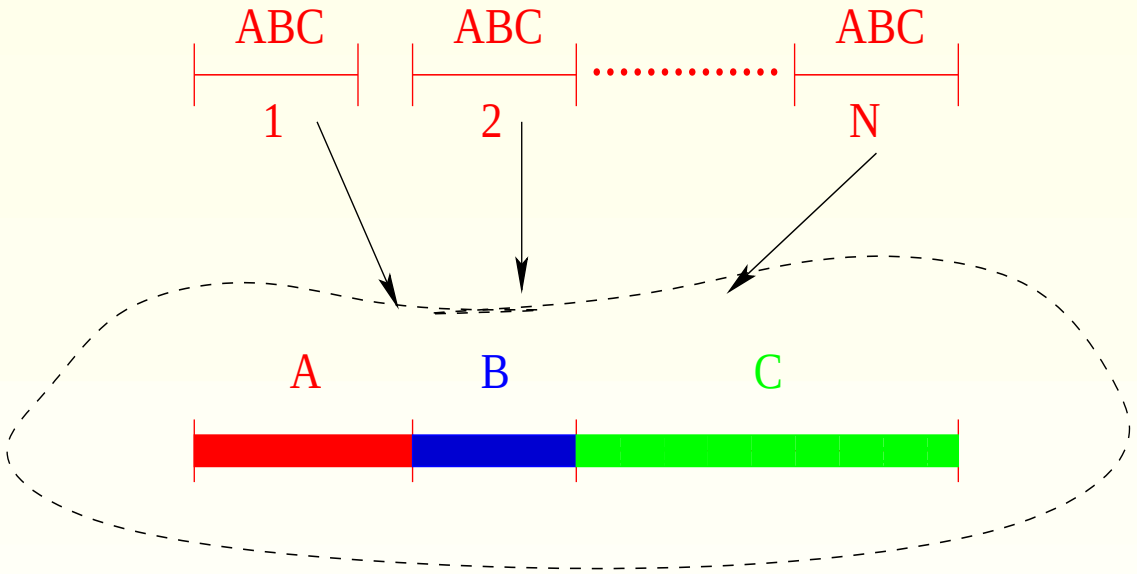
- Transition period: dynamic transitions between two products take place
- Production period: a given product is manufactured around steady-state conditions



Mixed-Inter Dynamic Optimization Formulation

Single Stage Cyclic Scheduling Formulation for Continuous Plants

In this section we will describe a scheduling formulation for continuous plants as proposed by Pinto and Grossmann. The formulation assumes a cyclic production sequence. Let us assume that a given plant manufactures the products A,B,C in the sequence A $\longrightarrow$ B $\longrightarrow$ C:



- Objective function

$$\phi = \sum_{i=1}^{N_p} \frac{C_i^p W_i}{T_c} - \sum_{i}^{N_p} \sum_{i'}^{N_p} \sum_{k}^{N_s} \frac{C^t_{i'\,i}\, Z_{ii'k}}{T_c} - \sum_{i=1}^{N_p} C_i^s \frac{(G_i - W_i/T_c)}{T_c} \frac{t_i}{2} \qquad (1)$$

where C_i^p is the product cost, $C^t_{i'\,i}$ is the transition cost from product i to product $i^{'}$, C_i^s is the inventory cost, T_c is the total cycle time, N_p is the number of products, N_s is the number of slots. The subscripts i and $i^{'}$ stand for products, whereas k denotes slot number.

- Product assignment

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

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

$$y'_{ik} = y_{i,k-1}, \ \forall i, k \neq 1 \tag{2c}$$

$$y'_{i,1} = y_{i,N_s}, \ \forall i \tag{2d}$$

Equation 2a states that, within each production wheel, any product can only be manufactured once, while constraint 2b implies that only one product is manufactured at each time slot. Due to this constraint, the number of products and slots turns out to be the same. Equation 2c defines backward binary variable (y'_{ik}) meaning that such variable for product i in slot k takes the value assigned to the same binary variable but one slot backwards $k-1$. At the first slot, Equation 2d defines the backward binary variable as the value of the same variable at the last slot. This type of assignment reflects our assumption of cyclic production wheel. The variable y'_{ik} will be used later to determine the sequence of product transitions.

- Amounts manufactured

$$W_i \geqslant D_i T_c, \ \forall i \tag{3a}$$

$$W_i = G_i \Theta_i, \ \forall i \tag{3b}$$

$$G_i = F^o(1 - X_i), \ \forall i \tag{3c}$$

Equation 3a states that the total amount manufactured of each product i (W_i) must be equal or greater than the desired demand rate (D_i) times the duration of the production wheel, while Equation 3b indicates that the amount manufactured of product i is computed as the product of the production rate (G_i) times the time used (Θ_i) for manufacturing such product. The production rate is computed from Equation 3c as a simple relationship between the feed stream flowrate (F^o) and the conversion (X_i).

- Processing times

$$\theta_{ik} \leqslant \theta^{max} y_{ik}, \ \forall i, k \tag{4a}$$

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

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

The constraint given by Equation 4a sets an upper bound on the time used for manufacturing product i at slot k (θ_{ik}). Equation 4b is the time used for manufacturing product i, while Equation 4c defines the duration time at slot k (p_k).

- Transitions between products

$$z_{ipk} \geqslant y'_{pk} + y_{ik} - 1, \ \forall i, p, k \tag{5}$$

The constraint given in Equation 5 is used for defining the binary production transition

variable z_{ipk}. If such variable is equal to 1 then a dynamic transition will occur from product i to product p within slot k, z_{ipk} will be zero otherwise.

- Timing relations

$$\theta_k^t = \sum_{i=1}^{N_p} \sum_{p=1}^{N_p} t_{pi}^t z_{ipk}, \ \forall k \tag{6a}$$

$$t_1^s = 0 \tag{6b}$$

$$t_k^e = t_k^s + p_k + \sum_{i=1}^{N_p} \sum_{p=1}^{N_p} t_{pi}^t z_{ipk}, \ \forall k \tag{6c}$$

$$t_k^s = t_{k-1}^e, \ \forall k \neq 1 \tag{6d}$$

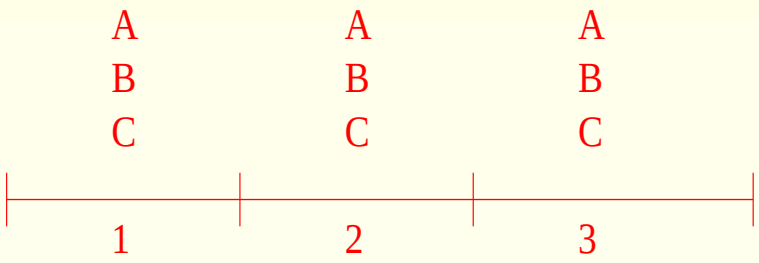
$$t_k^e \leqslant T_c, \ \forall k \tag{6e}$$

$$t_{fck} = (f-1)\frac{\theta_k^t}{N_{fe}} + \frac{\theta_k^t}{N_{fe}}\gamma_c, \ \forall f, c, k \tag{6f}$$

Equation 6a defines the transition time from product i to product p at slot k. It should be remarked that the term t_{pi}^t stands only for an estimate of the expected transition times. Equation 6b sets to zero the time at the beginning of the production wheel cycle corresponding to the first slot. Equation 6c is used for computing the time at the end of each slot as the sum of the slot start time plus the processing time and the transition time. Equation 6d states that the start time at all the slots, different than the first one, is just the end time of the previous slot. Equation 6e is used to force that the end time at each slot be less than the production wheel cyclic time. Finally, Equation 6f is used to obtain the time value inside each finite element and for each internal collocation point.

Example

Let us assume that a given plant facility manufactures three products: A,B,C. We would like to compute the optimal cyclic production sequence that maximizes the process profit while meeting the demand rate of each product.



Transition times
(Transition cost)

	A	B	C
A		10 (4000)	20 (8000)
B	15 (3500)		30 (6000)
C	10 (7000)	25 (5500)	

Process data

Product	Demand rate	Price Products	Process rates	Inventory cost
A	2.1	320	8	1.5
B	3	430	9	1
C	4	450	12	2

Optimal Cyclic Scheduling Results: Profit=329

Slot	Product	Process time	Amount Manufactured	Start time	End time
1	B	235.3	2117.7	0	265.3
2	A	185.3	1482.4	265.3	460.6
3	C	235.3	2823.5	460.6	705.9

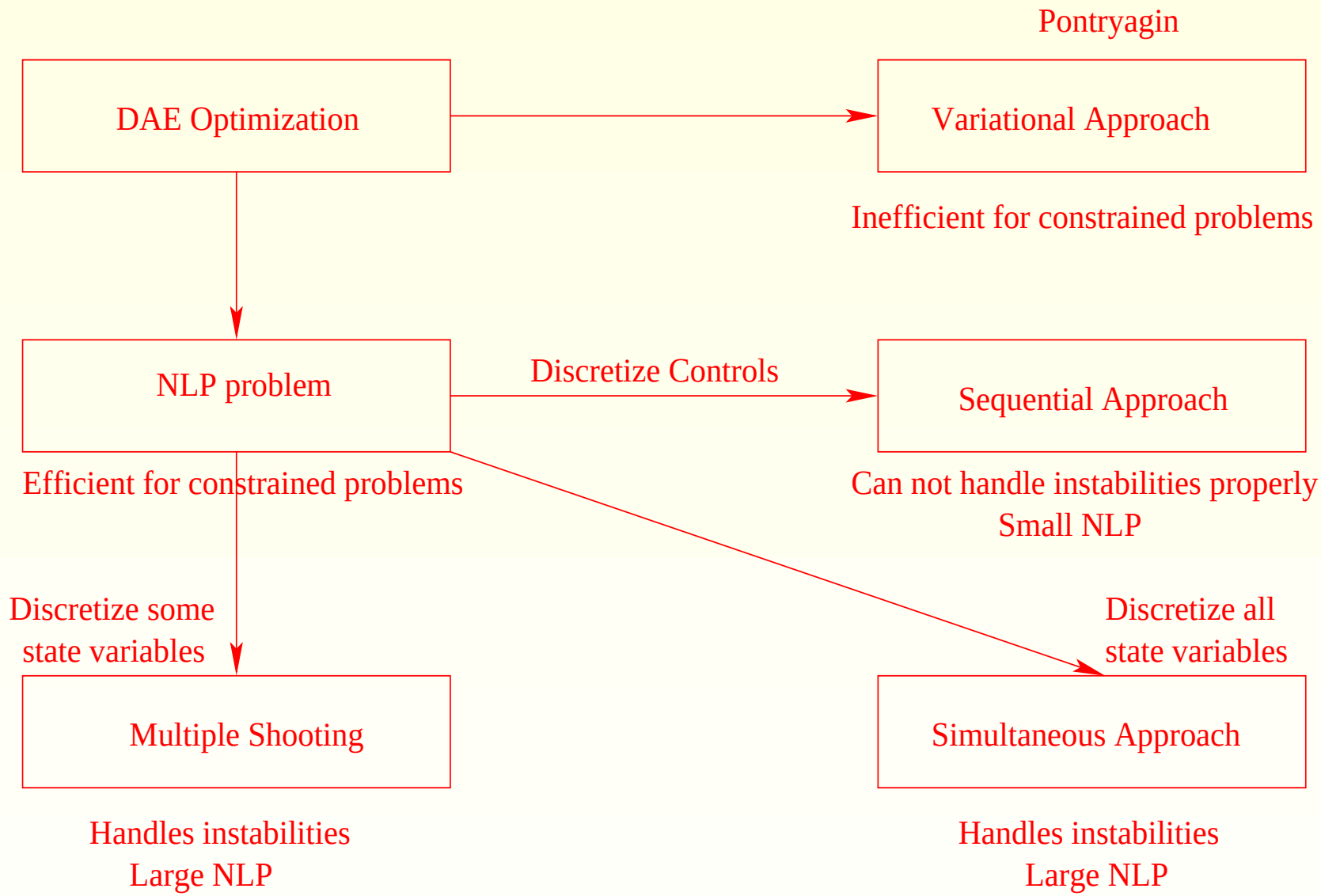
```
$title Single Stage Scheduling Problem

set      i Tasks / A, B, C /

         k Slots / 1*3 /
         kk      /2*3/
                 ;
alias(i,ip)              ;

table tt(ip,i) "transition times "
                 A       B       C
         A       0       10      20
         B       15      0       30
         C       10      25      0 ;

table ct(ip,i) "cost of transition"
                 A              B              C
         A       0              4000          8000
         B       3500            0             6000
         C       7000           5500            0 ;

parameter  d(i) demand rate
         /A 2.1, B 3, C 4/

           pr(i) price products
         /A 320, B 430, c 450 /

           g(i) process rates
```

```
        /A 8, B 9, C 12/

          ca(i) cost of inventory
        /A 1.5, B 1, C 2  /;

parameter u / 6000 / ;

variables profit    profit ;

positive variables  Tc      cycle time
                    t(i)    processing of prod i
                    w(i)    amount produced prod i in Tc
                    ts(k)   Start of slot l
                    te(k)   end of slot l
                    p(k)    process time in slot l
                    th(i,k) prod i in slot k ;

binary variables    y(i,k)  prod i in time slot k
                    yp(ip,k) additional y;

positive variables   z(i,ip,k) transition var  ;

equations         obj             profit function
                  oneSlot(i)       only one slot per task
                  oneTask(k)       only one prod per slot
                  endstartSlot(k)  balance end and start times
                  procTime(k)      process time is
                  supportEq(i,k)   support equation to determine the prod time of i in slot k
```

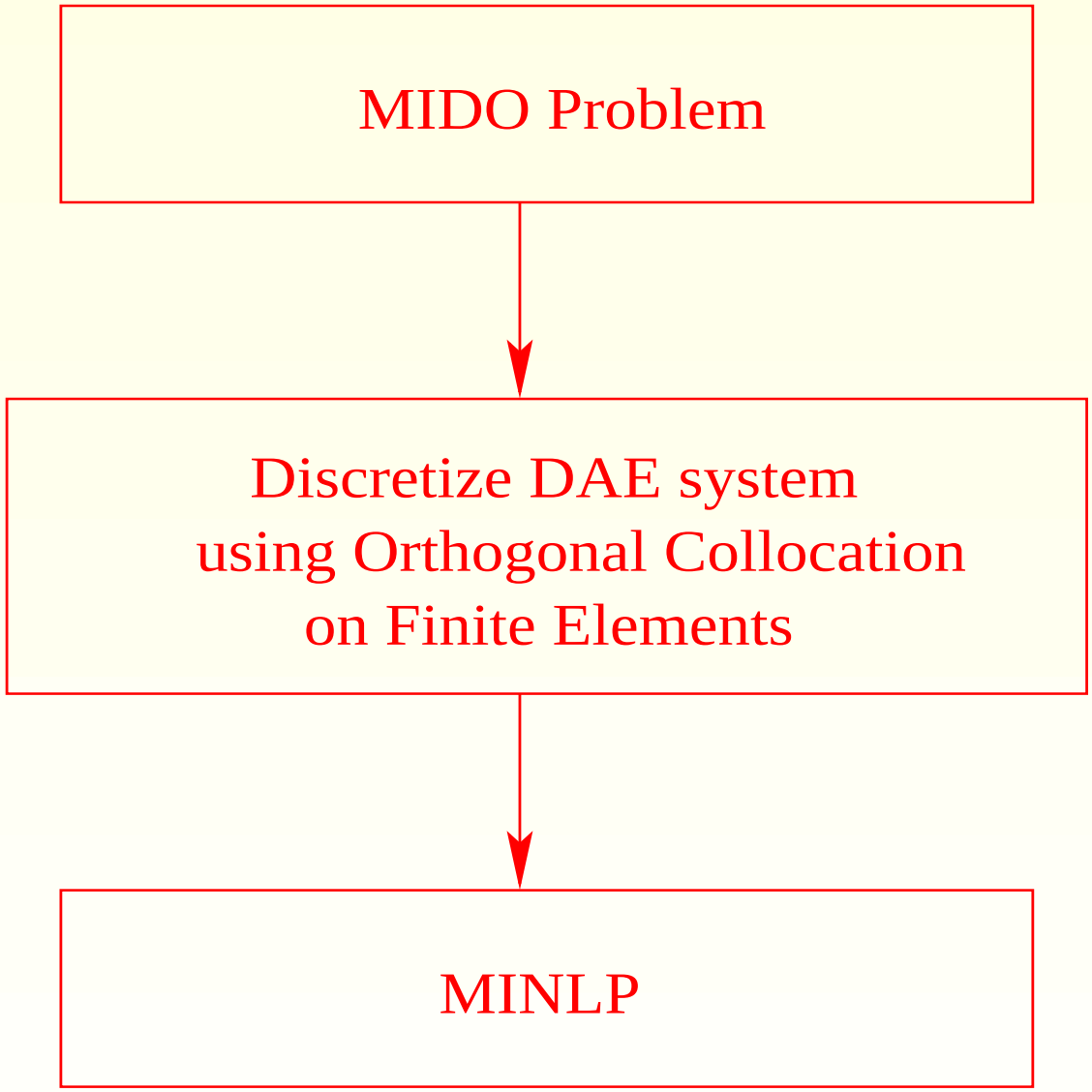
```
                switch(i,ip,k)   switching transition at slot k
                Productreq(i)    amount of i required in Tc
                Production(i)    production of i
                Processtime(i)   processing time of prod i
                endStartPN(k)    balance End and Start of successive slots
                Cyctime(k)       total cycle time
                startA           first ts
                assignyp(i,k)
                assignyp1(i)    ;

obj .. Profit =e= sum (i, pr(i)*W(i))/Tc- sum((i,ip,k), ct(ip,i)*z(i,ip,k)/ Tc)
                 - sum(i, ca(i) * ((g(i) - w(i)/Tc) * t(i) / 2));

assignyp(i,k)$kk(k) .. yp(i,k)   =e= y(i,k-1);
assignyp1(i) ..        yp(i,'1') =e= y(i,'3');

switch(i,ip,k) .. z(i,ip,k) =g= yp(ip,k) + y(i,k) - 1 ;

oneSlot(i) .. sum(k,y(i,k)) =e= 1 ;

oneTask(k) .. sum(i,y(i,k)) =e= 1 ;

endstartSlot(k) ..  te(k) =e= ts(k) + p(k) + sum((i,ip), tt(i,ip) * z(i,ip,k)) ;

procTime(k) .. p(k) =e= sum(i,th(i,k)) ;

supportEq(i,k) .. th(i,k) =l= y(i,k) *  u ;
```

```
Productreq(i) .. w(i) =g= d(i) * Tc ;

Production(i) .. w(i) =e= g(i) * t(i) ;

Processtime(i) .. t(i) =e= sum(k, th(i,k)) ;

endStartPN(k)$kk(k) .. ts(k) =e= te(k-1) ;

Cyctime(k) .. te(k) =l= Tc ;

startA .. ts('1') =e= 0 ;

Tc.l = .2;
Tc.lo=1;
Tc.UP=6000;

options limrow = 0;
options limcol = 0;

model mySched / all / ;

solve mySched maximizing profit using minlp ;
```

Dynamic Optimization Approaches



MIDO Simultaneous Approach



Discretizing ODEs using Orthogonal Collocation

Given an ODE system:

$$\frac{dx}{dt} = f(x, u, p), \quad x(0) = x_{init}$$

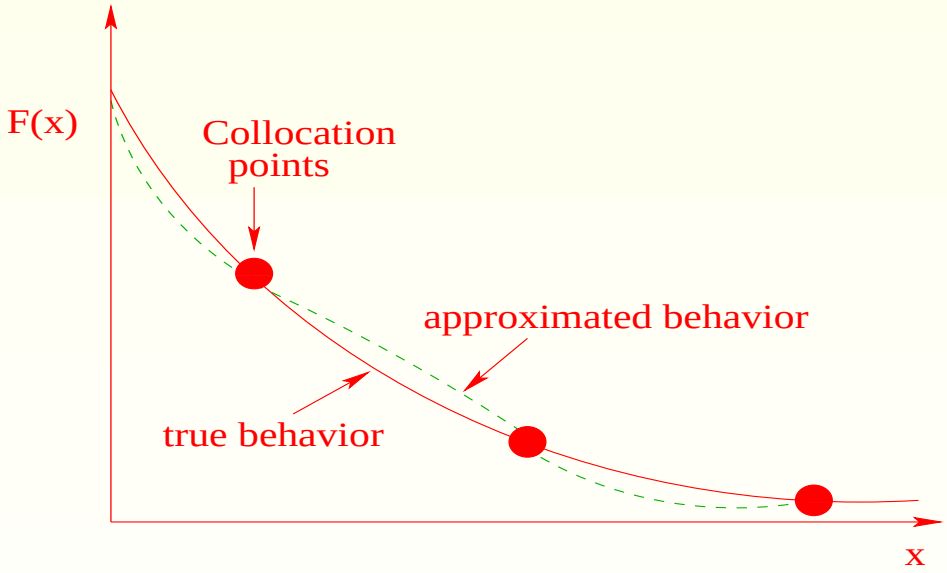
where $x(t)$ are the system states, $u(t)$ is the manipulated variable and p are the system parameters.

The aim is to approximate the behaviour of x and u by Lagrange interpolation polynomials (of orders $\mathcal{K}+1$ and $\mathcal{K}$, respectively) at collocation or discretization points t_k:

$$x_{k+1}(t) = \sum_{k=0}^{\mathcal{K}} x_k \ell_k^x(t), \quad \ell_k^x(t) = \prod_{\substack{j=0 \\ j\neq k}}^{\mathcal{K}} \frac{t - t_j}{t_k - t_j}$$

$$u_k(t) = \sum_{k=1}^{\mathcal{K}} u_k \ell_k^u(t), \quad \ell_k^u(t) = \prod_{\substack{j=1 \\ j\neq k}}^{\mathcal{K}} \frac{t - t_j}{t_k - t_j}$$

$$x_{N+1}(t_k) = x_k, \quad u_N(t_k) = u_k$$



Therefore replacing into the original ODE system, we get the system residual $\mathcal{R}(t_k)$:

$$\mathcal{R}(t_k) = \sum_{j=0}^{\mathcal{K}} x_j \frac{d\ell_j(t_k)}{dt} - f(x_k, u_k, p) = 0, \quad k = 1, .., \mathcal{K}$$

Transformation of a Dynamic Optimization problem into a NLP

Original dynamic optimization problem

$$\min_{x,u} \phi(x,u)$$

s.t.
$$\frac{dx(t)}{dt} = F\left(x(t),u(t),t,p\right)$$
$$x(0) = x^0$$
$$g(x(t),u(t),p) \leqslant 0$$
$$h(x(t),u(t),p) = 0$$
$$x^l \leqslant x \leqslant x^u$$
$$u^l \leqslant u \leqslant u^u$$

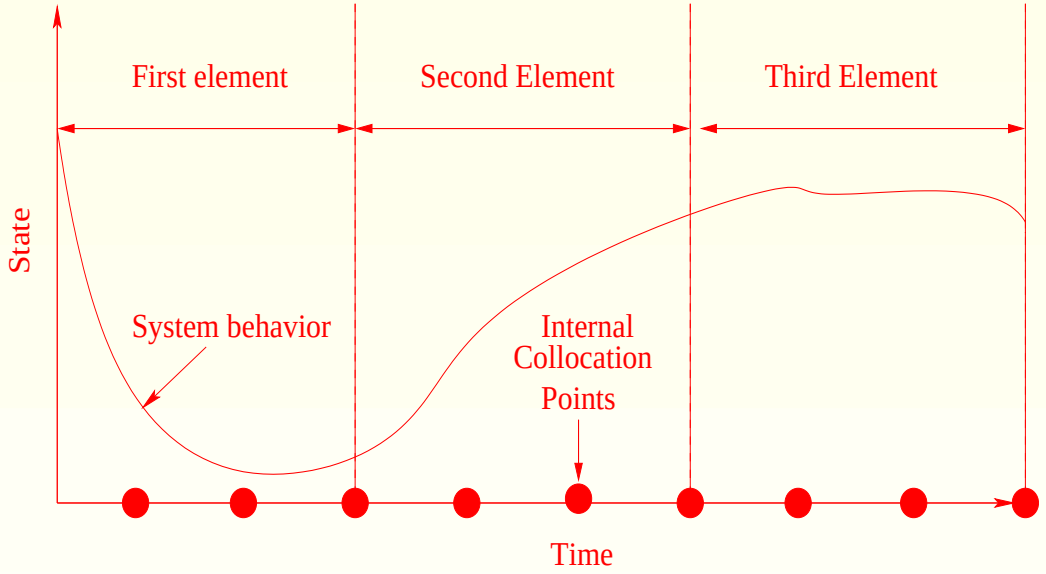
Discretized NLP

$$\min_{x_k,u_k} \phi(x_k,u_k)$$

s.t.
$$\sum_{j=0}^{\mathcal{K}} x_j \dot{\ell}_j(t_k) - F\left(x_k,u_k\right) = 0,\ k = 1,\ldots,\mathcal{K}$$
$$x_0 = x(0)$$
$$g(x_k,u_k,p) \leqslant 0,\ k = 1,\ldots,\mathcal{K}$$
$$h(x_k,u_k,p) = 0,\ k = 1,\ldots,\mathcal{K}$$
$$x^l \leqslant x \leqslant x^u$$
$$u^l \leqslant u \leqslant u^u$$

Approximation of a Dynamic Optimization Problem using Orthogonal Collocation of Finite Elements

Sometimes it is convenient to use Orthogonal Collocation on Finite Elements to approximate the behavior of systems exhibiting fast dynamics.



$$\min_{x_k, u_k} \phi(x, u)$$

s.t.

$$\sum_{j=0}^{\mathcal{K}} x_{ij} \dot{\ell}_j(\tau_k) - h_i F(x_{ik}, u_{ik}) = 0, \quad k = 1, \ldots, NC \quad {}^{i=1,\ldots,NE}$$

$$x_{10} = x(0)$$

$$g(x_{ik}, u_{ik}, p) = 0, \quad i = 1, \ldots, NE; k = 1, .., NC$$

$$x_{ij}^{l} \leqslant x_{ij} \leqslant x_{ij}^{u}, \quad i = 1, \ldots, NE; k = 1, .., NC$$

$$u_{ij}^{l} \leqslant u_{ij} \leqslant u_{ij}^{u}, \quad i = 1, \ldots, NE; k = 1, .., NC$$

where NE is the number of finite elements, NC is the number of internal collocation points, h_i is the length of each element.

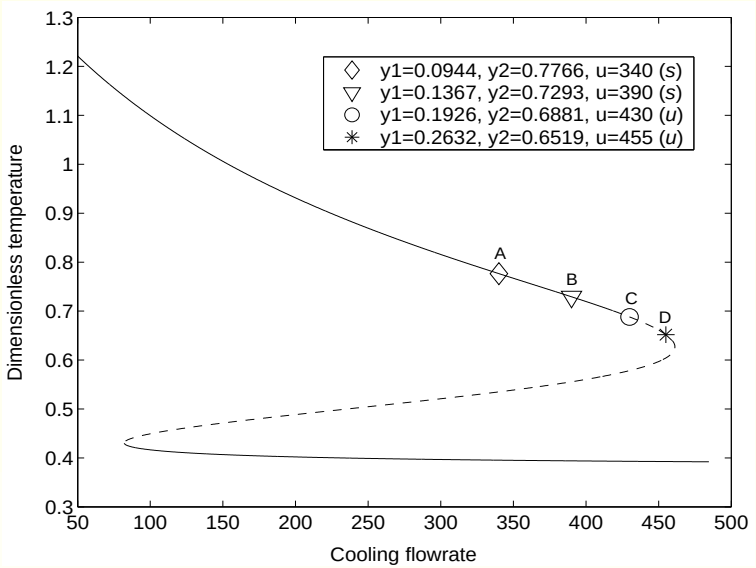
Dynamic optimal transition between two steady-states: Hicks CSTR

Let us consider the dimensionless mathematical model of a non-isothermal CSTR as proposed by Hicks and Ray modified for displaying nonlinearities:

$$\frac{dC}{dt} = \frac{1 - C}{\theta} - k_{10} e^{-N/T} C$$

$$\frac{dT}{dt} = \frac{y_f - T}{\theta} + k_{10} e^{-N/T} C - \alpha U (T - y_c)$$

Desired Transition B $\rightarrow$ A



Parameter values

θ	20	Residence time
T_f	300	Feed temperature
J	100	$(-\Delta H)/(\rho C_p)$
k_{10}	300	Preexponential factor
c_f	7.6	Feed concentration
T_c	290	Coolant temperature
α	1.95×10^{-4}	Heat transfer area
N	5	$E_1/(RJc_f)$

Desired dynamic transition

	C	T	U
Initial (B)	0.1367	0.7293	390
Final (A)	0.0944	0.7766	340

C = Concentration (c/c_f), T = temperature (T_r/Jc_f), y_c = Coolant temperature (T_c/Jc_f), y_f = feed temperature (T_f/Jc_f), U = Cooling flowrate. c and T_r are nondimensionless concentration and reactor temperature.

For solving this example we will use three finite elements and two internal collocation points.

- Objective function

  As objective function the requirement of minimum transition time between the initial and final steady-states will be imposed:

$$\text{Min} \int_0^{t_f} \left\{ \alpha_1 (C(t) - C_{des})^2 + \alpha_2 (T(t) - T_{des})^2 + \alpha_3 (U(t) - U_{des})^2 \right\} dt$$

  the subscript "des" stands for the final desired values. α_i, $i = 1, 2, 3$ are weighting factors. The above integral is approximated using a Radau quadrature procedure:

$$\text{Min } \Phi = \sum_{i=1}^{N_e} h_i \sum_{j=1}^{N_C} W_j \left[\alpha_1 (C_{ij} - C_{des})^2 + \alpha_2 (T_{ij} - T_{des})^2 + \alpha_3 (U_{ij} - U_{des})^2 \right]$$

  N_e is the number of finite elements (N_e=3), N_c is the number of collocation points including the right boundary in each element (so in this case $N_c = 3$), C_{ij} and T_{ij} are the dimensionless concentration and temperature values at each discretized ij point, h_i is the finite element length of the $i-$th element, W_j are the Radau quadrature weights.

- Constraints

  1. Mass balance

     The value of the dimensionless concentration at each one of the discretized points (C_{ij}) is approximated using the following monomial basis representation:

$$C_{ij} = C_i^o + h_i \theta \sum_{k=1}^{N_C} A_{kj} \frac{dC_{ik}}{dt}, \quad i = 1, \ldots, N_e; \quad j = 1, \ldots, N_c$$

C_i^o is the concentration at the beginning of each element, A_{kj} is the collocation matrix. Note that C_1^o stands for the initial concentration. The length of each finite element (h_i)can be computed as:

$$h_i = \frac{1}{N_e}$$

2. Energy balance

$$T_{ij} = T_i^o + h_i \theta \sum_{k=1}^{N_C} A_{kj} \frac{dT_{ik}}{dt}, \quad i = 1, \ldots, N_e; \quad j = 1, \ldots, N_c$$

similarly, T_i^o is the temperature at the beginning of each element. Again, note that T_1^o stands for the initial reactor temperature.

3. Mass balance continuity constrains between finite elements

Only the system states must be continuous when crossing from one finite element to the next one. Algebraic and manipulated variables are allowed to exhibit discontinuous behaviour between finite elements. To force continuous concentration profiles all the elements at the beginning of each element ($C_i, \ i = 2, \ldots, N_e^o$) are computed in terms of the same monomial basis used before:

$$C_i^o = C_{i-1}^o + h_{i-1} \theta \sum_{k=1}^{N_C} A_{k,N_C} \frac{dC_{i-1,k}}{dt}, \quad i = 2, \ldots, N_e$$

4. Energy balance continuity constrains between finite elements

$$T_i^o = T_{i-1}^o + h_{i-1} \theta \sum_{k=1}^{N_C} A_{k,N_C} \frac{dT_{i-1,k}}{dt}, \quad i = 2, \ldots, N_e$$

5. Approximation of the dynamic behaviour of the mass balance at each collocation point

   The first order derivatives of the concentration at each collocation point (ij) are obtained from the corresponding continuous mathematical model:

$$\frac{dC_{i,j}}{dt} = \frac{1 - C_{ij}}{\theta} - k_{10} e^{-N/T_{ij}} C_{ij}, \quad i = 1, \ldots, N_e; \quad j = 1, \ldots, N_c$$

6. Approximation of the dynamic behaviour of the energy balance at each collocation point

$$\frac{dT_{i,j}}{dt} = \frac{y_f - T_{ij}}{\theta} + k_{10} e^{-N/T_{ij}} C_{ij} - \alpha U_{ij} (T_{ij} - y_c), \quad i = 1, \ldots, N_e; \quad j = 1, \ldots, N_c$$

7. Initial values constraints

$$\begin{aligned} C_1^o &= C_{init} \\ T_1^o &= T_{init} \end{aligned}$$

   the subscript "init" stands for the initial steady-state values from which the optimal dynamic transition will be computed.

The collocation matrix for 2 internal points is given as follows

$$A = \begin{bmatrix} 0.19681547722366 & 0.39442431473909 & 0.37640306270047 \\ -0.06553542585020 & 0.29207341166523 & 0.51248582618842 \\ 0.02377097434822 & -0.04154875212600 & 0.11111111111111 \end{bmatrix}$$

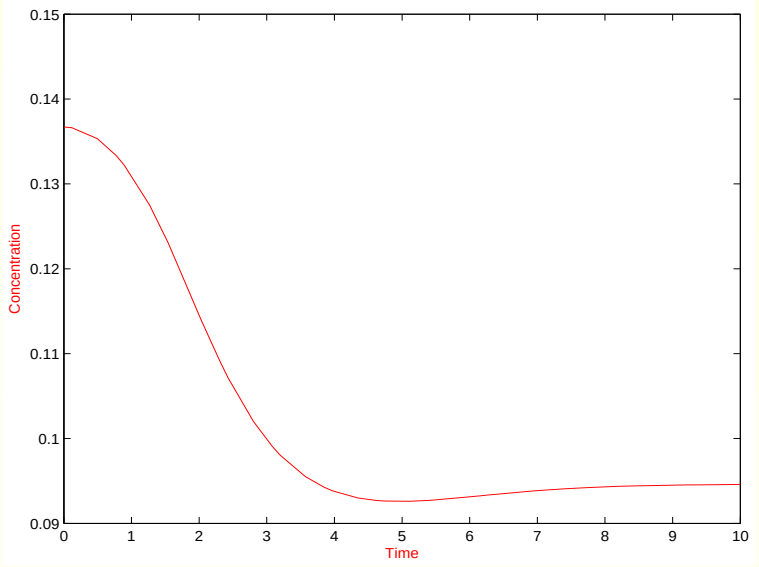
while the Radau cuadrature weights are

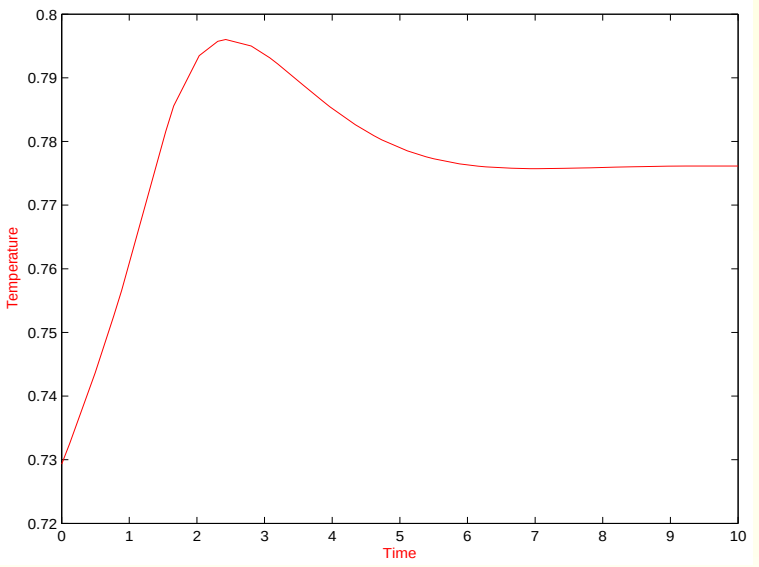
$$W = \begin{bmatrix} 0.37640306270047 \\ 0.51248582618842 \\ 0.11111111111111 \end{bmatrix}$$

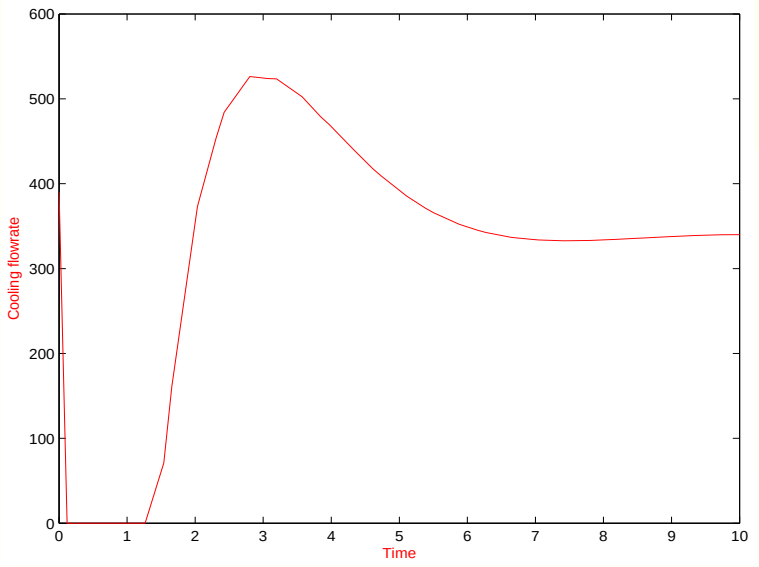
the roots (R) of the interpolating polynomial are needed for descaling the time variable.

$$R = \begin{bmatrix} 0.1550625 \\ 0.6449948 \\ 1 \end{bmatrix}$$

Dynamic Transitions profiles for the Hicks CSTR example







```
#
# Dynamic optimization of the Hicks CSTR problem
#
# Written by Antonio Flores T./CMU, 31 Jan, 2004
#
param NFE >= 1 integer       ; # Number of finite elements
param NCP >= 1, <= 5 integer ; # Number of collocation points
#
# Define initial values and final desired ones
#
param Cinit     >= 0   ;
param Tinit     >= 0   ;
param Uinit     >= 0   ;
param Cdes      >= 0   ;
param Tdes      >= 0   ;
param Udes      >= 0   ;
param TransTime >= 0   ;
#
# Define specific parameters for Hicks multiplicity CSTR
#
param alpha  >= 0  ;
param alpha1 >= 0  ;
param alpha2 >= 0  ;
param alpha3 >= 0  ;
param k10    >= 0  ;
param N      >= 0  ;
param Cf     > 0   ;
```

```
param J      > 0   ;
param Tf     > 0   ;
param Tc     > 0   ;
param yf     >= 0  ;
param yc     >= 0  ;
param theta  > 0   ;
param r1     >= 0  ;
param r2     >= 0  ;
param r3     >= 0  ;
#
# Parameters for defining decision variables initial guesses
#
param POINT  ;
param SLOPEc ;
param SLOPEt ;
param SLOPEu ;
#
# Define dimensions for all indexed variables
#
set FE := 1..NFE ;
set CP := 1..NCP ;
param A {CP,CP}  ; # Collocation matrix
param H{FE};
#
# Define derivatives of the states evaluated at each collocation point
#
var Cdot {FE,CP} ;
var Tdot {FE,CP} ;
```

```
var TIME >= 0    ;
#
# Define the states value at the beginning of each finite element
#
var C0 {FE} >= 0.01, <= 1    ;
var T0 {FE} >= 0.01, <= 5    ;
var U0 {FE} >= 0, <= 2500    ;
#
# Define decision variables
#
var C {FE,CP} >= 0, <= 1    ; # Dimensionless concentration profile
var T {FE,CP} >= 0, <= 5    ; # Dimensionless temperature profile var U {FE,CP} >= 0, <= 25
# Objective function
#
minimize COST:
sum{i in FE} (H[i]*sum{j in CP} ((
alpha1*(C[i,j]-Cdes)^2+alpha2*(T[i,j]-Tdes)^2+alpha3*(U[i,j]-Udes)^2)*A[j,NCP]));
#
# Mass and Energy balance discretization
#
subject to FECOLc{i in FE,j in CP}:
C[i,j] = C0[i]+TIME*H[i]*sum{k in CP} A[k,j]*Cdot[i,k];

subject to FECOLt{i in FE,j in CP}:
T[i,j] = T0[i]+TIME*H[i]*sum{k in CP} A[k,j]*Tdot[i,k];
#
# Mass and Energy continuity constraints between finite elements
#
```

```
subject to CONc{i in FE diff{1}} :
C0[i] = C0[i-1]+TIME*H[i-1]*sum{k in CP} A[k,NCP]*Cdot[i-1,k] ;

subject to CONt{i in FE diff{1}} :
T0[i] = T0[i-1]+TIME*H[i-1]*sum{k in CP} A[k,NCP]*Tdot[i-1,k] ;
#
# Approximation of the Mass and Energy derivatives at each collocation point
#
subject to ODEc{i in FE, j in CP} :
Cdot[i,j] = (1-C[i,j])/theta-k10*exp(-N/T[i,j])*C[i,j] ;

subject to ODEt{i in FE, j in CP} :
Tdot[i,j] = (yf-T[i,j])/theta+k10*exp(-N/T[i,j])*C[i,j]-alpha*U[i,j]*(T[i,j]-yc) ;
#
# Initial conditions constraints
#
subject to IVc: C0[1] = Cinit ;
subject to IVt: T0[1] = Tinit ;
subject to IVu: U0[1] = Uinit ;
#
# Constraint on the total transition time
#
subject to TTT: TIME = TransTime ;

# -- End of the hicks.mod file --
```

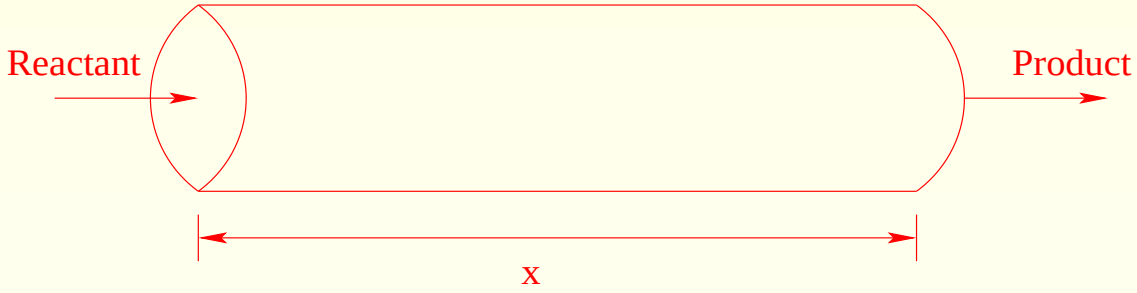
```
#
# This file contains all the information to run one
# of the cases of the Hicks dynamic optimization problem
#
#
# First order derivatives collocation matrix
#
param  A:        1                   2                   3 :=
      1  0.19681547722366   0.39442431473909   0.37640306270047
      2 -0.06553542585020   0.29207341166523   0.51248582618842
      3  0.02377097434822  -0.04154875212600   0.11111111111111;

let NFE       := 13 ;
let NCP       := 3  ;
let TransTime := 10 ;

let r1 := 0.15505102572168 ;
let r2 := 0.64494897427832 ;
let r3 :=  1               ;
#
# Initial value fixed conditions and final (desired) conditions
#
let Cinit  := 0.1367 ;
let Tinit  := 0.7293 ;
let Uinit  := 390    ;
let Cdes   := 0.0944 ;
let Tdes   := 0.7766 ;
```

```
let Udes   := 340    ;
#
# CSTR parameters (modified for multiplicity behaviour)
#
let alpha  := 1.95e-04  ;
let alpha1 := 1e+06     ;
let alpha2 := 2e+03     ;
let alpha3 := 1e-03     ;
let k10    := 300       ;
let N      := 5         ;
let Cf     := 7.6       ;
let J      := 100       ;
let Tf     := 300       ;
let Tc     := 290       ;
let theta  := 20        ;
let yf     := Tf/(J*Cf) ;
let yc     := Tc/(J*Cf) ;
#
# In this section initial guesses of the decision variables are
# computed. They consists on simple linear interpolations between
# the initial fixed values and the desired ones.
#
let POINT  := 0                       ;
let SLOPEc := (Cdes-Cinit)/(NFE*NCP) ;
let SLOPEt := (Tdes-Tinit)/(NFE*NCP) ;
let SLOPEu := (Udes-Uinit)/(NFE*NCP) ;

for {i in FE}
{
```

```
    for {j in CP}
    {
         let POINT := POINT+1;
         let C[i,j] := SLOPEc*POINT+Cinit ;
         let T[i,j] := SLOPEt*POINT+Tinit ;
         let U[i,j] := SLOPEu*POINT+Uinit ;
     } let H[i] := 1/NFE ;
}

#-- End of the hicks.dat file --
```

Simultaneous Dynamic Optimization Example: PDE Optimization

Let us consider the dynamic optimization of a distributed parameter system. Specifically we will deal with the mathematical model a of dynamic, one dimensional isothermal tubular reactor with diffusive and convective mass transfer:



Mathematical model

$$\begin{aligned} \frac{\partial c}{\partial t} &= \frac{\partial^2 c}{\partial x^2} - Pe_M \frac{\partial c}{\partial x} - Pe_M R(c) \\ R(c) &= \alpha K c^2 \end{aligned}$$

subject to the following initial,

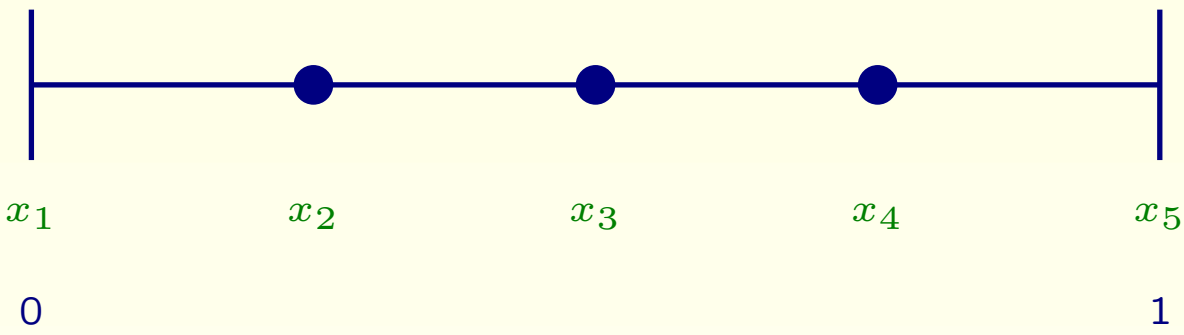
$$c(x, 0) = 1$$

and boundary conditions,

$$\begin{aligned} \frac{\partial c}{\partial x} &= Pe_M(c - 1), \ @\ x = 0 \\ \frac{\partial c}{\partial x} &= 0 \qquad\qquad , \ @\ x = 1 \end{aligned}$$

where c is the dimensionless concentration, x is the dimensionless axial coordinate, Pe_M is the mass Peclet number, K is the cinetic rate constant, α is a constant, and t is the time.

In this example we will use only three internal collocation points as depicted below.



approximating the first and second order spatial derivatives at each i internal collocation point,

$$\left(\frac{\partial c}{\partial x}\right)_i = \sum_{j=1}^{N+2} A_{ij} c_j, \quad \left(\frac{\partial^2 c}{\partial x^2}\right)_i = \sum_{j=1}^{N+2} B_{ij} c_j$$

Therefore, if we discretize the mathematical model,

$$\left(\frac{\partial c}{\partial x}\right)_i = \sum_{j=1}^{N+2} B_{ij} c_j - Pe_M \sum_{j=1}^{N+2} A_{ij} c_j - Pe_M R(c_i), \; i = 2, .., N+1$$

and the boundary conditions,

$$\begin{aligned}
\frac{\partial c}{\partial x} &= \sum_{j=1}^{N+2} A_{1j} c_j - Pe_M (c_1 - 1), \quad @\ x = 0 \\
\frac{\partial c}{\partial x} &= \sum_{j=1}^{N+2} A_{ij} c_j = 0, \qquad\qquad @\ x = 1
\end{aligned}$$

If we now expand the above equations using three internal collocation points,

$$\begin{aligned}
0 &= A_{11}c_1 + A_{12}c_2 + A_{13}c_3 + A_{14}c_4 + A_{15}c_5 - Pe_M(c_1 - 1) \\
\frac{\partial c_2}{\partial t} &= B_{21}c_1 + B_{22}c_2 + B_{23}c_3 + B_{24}c_4 + B_{25}c_5 - Pe_M[A_{21}c_1 + A_{22}c_2 \\
&\quad + A_{23}c_3 + A_{24}c_4 + A_{25}c_5] - Pe_M \alpha K c_2^2 \\
\frac{\partial c_3}{\partial t} &= B_{31}c_1 + B_{32}c_2 + B_{33}c_3 + B_{34}c_4 + B_{35}c_5 - Pe_M[A_{31}c_1 + A_{32}c_2 \\
&\quad + A_{33}c_3 + A_{34}c_4 + A_{35}c_5] - Pe_M \alpha K c_3^2 \\
\frac{\partial c_4}{\partial t} &= B_{41}c_1 + B_{42}c_2 + B_{43}c_3 + B_{44}c_4 + B_{45}c_5 - Pe_M[A_{41}c_1 + A_{42}c_2 \\
&\quad + A_{43}c_3 + A_{44}c_4 + A_{45}c_5] - Pe_M \alpha K c_4^2 \\
0 &= A_{51}c_1 + A_{52}c_2 + A_{53}c_3 + A_{54}c_4 + A_{55}c_5
\end{aligned}$$

In this note we will use Collocation matrices based on Lagrange polynomials to approximate the

first and second order spatial derivatives:

```
A =

  -13.0000   14.7883   -2.6667    1.8784   -1.0000
   -5.3238    3.8730    2.0656   -1.2910    0.6762
    1.5000   -3.2275    0.0000    3.2275   -1.5000
   -0.6762    1.2910   -2.0656   -3.8730    5.3238
    1.0000   -1.8784    2.6667  -14.7883   13.0000


B =

   84.0000 -122.0632   58.6667  -44.6035   24.0000
   53.2379  -73.3333   26.6667  -13.3333    6.7621
   -6.0000   16.6667  -21.3333   16.6667   -6.0000
    6.7621  -13.3333   26.6667  -73.3333   53.2379
   24.0000  -44.6035   58.6667 -122.0632   84.0000
```

The time derivative will be approximated using an implicit Runge-Kutta method. For time approximation two internal collocation points will be used. As objective function we will pose the following function featuring minimum transition time between two arbitrary operating points:

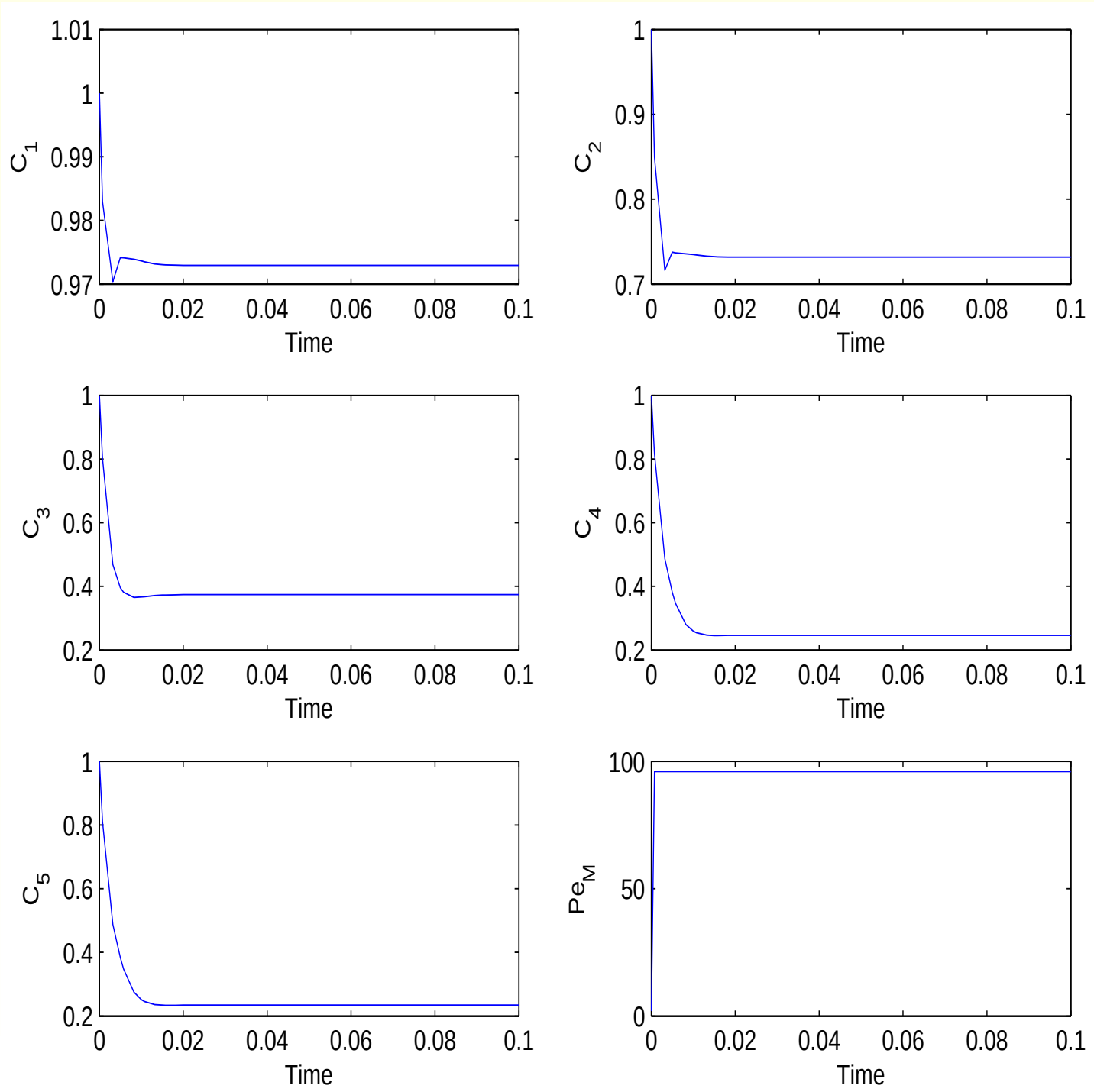
$$\text{Min } \Phi = \int_0^{t_f} \left\{ \left(C_5(t) - \hat{C}_5 \right)^2 + \left(Pe_M(t) - \hat{Pe}_M \right)^2 \right\} dt$$

The above equation states that we would like to move from an initial point to a final desired exit product concentration (denoted by $\hat{C}_5$) using the mass Peclet number (Pe_M) as the manipulated variable. The final transition value of the Peclet number is denoted by $\hat{Pe}_M$. In this example we will compute an optimal dynamic transition between the operating conditions shown in the next Table.

Desired dynamic transition

	C_5	Pe_M
Initial	1	2
Final	0.13	96

Dynamic optimization results for a tubular reactor with diffusive and convective mass transfer



Lagrange Collocation Matrices

```
clear all; clc;
%
% Program to compute the A,B (first and second order derivarives of
% Lagrange Polynomials) at the locations given by 'roots'.
%
% Written by Antonio Flores T./ 4 March, 2008
%
N=4;
roots=[                  0
    1.127016653792584e-001
    4.999999999999999e-001
    8.872983346207419e-001
    1.000000000000000e+000];

syms x x0 x1 x2 x3 x4
syms num den
xvect = [x0 x1 x2 x3 x4];

for i = 1:N+1,
    num = 1;
    den = 1;
    for j = 1:N+1,
        if j ~= i
           num = num*(x-xvect(j));
           den = den*(xvect(i)-xvect(j));
        end
    end
```

```
    L (i) = num/den;
    Lp (i) = diff(L(i),'x');
    Lpp (i) = diff(Lp(i),'x');
end

x0 = roots(1);
x1 = roots(2);
x2 = roots(3);
x3 = roots(4);
x4 = roots(5);

for i = 1:N+1,
    x = roots(i);
    for j = 1:N+1,
        A(i,j) = subs(Lp(j));
        B(i,j) = subs(Lpp(j));
    end
end

%-- End of file --
```

AMPL files for Dynamic Optimization

```
#
# Dynamic optimization of a tubular reactor with
# Diffusive and Convective Mass Transfer
#
# Written by Antonio Flores T., 5 March, 2008
#
param NFE >= 1 integer       ; # Number of finite elements
param NCP >= 1, <= 5 integer ; # Number of collocation points
param NPOC >=1, <= 10 integer; # Number of collocation points for discretizing spatial derivatives
#
# Define initial values and final desired ones
#
param c2init    >= 0  ;
param c3init    >= 0  ;
param c4init    >= 0  ;
param c5des     >= 0  ;
param peminit   >= 0  ;
param pemdes    >= 0  ;
param TransTime >= 0  ;
param r1        >= 0  ;
param r2        >= 0  ;
param r3        >= 0  ;
#
# Define specific parameters
#
param alpha >= 0    ;
param alphac5 >= 0  ;
```

```
param alphapem >= 0 ;
param gamma >= 0    ;
param krate >= 0    ;
#
# Define dimensions for all indexed variables
#
set FE   := 1..NFE  ;
set CP   := 1..NCP  ;
set POC  := 1..NPOC ;
param A  {CP,CP}    ; # IRK matrix
param AL {POC,POC}  ; # Lagrange collocation matrix (first order spatial derivatives)
param BL {POC,POC}  ; # Lagrange collocation matrix (second order spatial derivatives)
param H  {FE}       ;
#
# Define derivatives of the states evaluated at each collocation point
#
var c2dot {FE,CP} ;
var c3dot {FE,CP} ;
var c4dot {FE,CP} ;
var TIME >= 0    ;
#
# Define the states value at the beginning of each finite element
#
var c02 {FE} >=0, <= 2;
var c03 {FE} >=0, <= 2;
var c04 {FE} >=0, <= 2;
#
# Define decision variables
#
```

```
var c1  {FE,CP} >=0, <= 2   ;
var c2  {FE,CP} >=0, <= 2   ;
var c3  {FE,CP} >=0, <= 2   ;
var c4  {FE,CP} >=0, <= 2   ;
var c5  {FE,CP} >=0, <= 2   ;
var pem {FE,CP} >=0, <= 150 ;
#
# Objective function
#
minimize COST:
sum{i in FE} (H[i]*sum{j in CP} ((
alphac5*(c5[i,j]-c5des)^2+alphapem*(pem[i,j]-pemdes)^2)*A[j,NCP]));
#
# Mass balance discretization
#
subject to FECOLc2{i in FE,j in CP}:
c2[i,j] = c02[i]+TIME*H[i]*sum{k in CP} A[k,j]*c2dot[i,k];

subject to FECOLc3{i in FE,j in CP}:
c3[i,j] = c03[i]+TIME*H[i]*sum{k in CP} A[k,j]*c3dot[i,k];

subject to FECOLc4{i in FE,j in CP}:
c4[i,j] = c04[i]+TIME*H[i]*sum{k in CP} A[k,j]*c4dot[i,k];
#
# Mass continuity constraints between finite elements
#
subject to CONc2{i in FE diff{1}} :
c02[i] = c02[i-1]+TIME*H[i-1]*sum{k in CP} A[k,NCP]*c2dot[i-1,k] ;
```

```
subject to CONc3{i in FE diff{1}} :
c03[i] = c03[i-1]+TIME*H[i-1]*sum{k in CP} A[k,NCP]*c3dot[i-1,k] ;

subject to CONc4{i in FE diff{1}} :
c04[i] = c04[i-1]+TIME*H[i-1]*sum{k in CP} A[k,NCP]*c4dot[i-1,k] ;
#
# Approximation of the Mass and Energy derivatives at each collocation point
#
subject to ODEc2{i in FE, j in CP} :
c2dot[i,j] = BL[2,1]*c1[i,j]+BL[2,2]*c2[i,j]+BL[2,3]*c3[i,j]+BL[2,4]*c4[i,j]+BL[2,5]*c5[i,j]
             -pem[i,j]*(AL[2,1]*c1[i,j]+AL[2,2]*c2[i,j]+AL[2,3]*c3[i,j]+AL[2,4]*c4[i,j]+AL[2,5]*c5
             -pem[i,j]*alpha*krate*c2[i,j]^2 ;

subject to ODEc3{i in FE, j in CP} :
c3dot[i,j] = BL[3,1]*c1[i,j]+BL[3,2]*c2[i,j]+BL[3,3]*c3[i,j]+BL[3,4]*c4[i,j]+BL[3,5]*c5[i,j]
             -pem[i,j]*(AL[3,1]*c1[i,j]+AL[3,2]*c2[i,j]+AL[3,3]*c3[i,j]+AL[3,4]*c4[i,j]+AL[3,5]*c5
             -pem[i,j]*alpha*krate*c3[i,j]^2 ;

subject to ODEc4{i in FE, j in CP} :
c4dot[i,j] = BL[4,1]*c1[i,j]+BL[4,2]*c2[i,j]+BL[4,3]*c3[i,j]+BL[4,4]*c4[i,j]+BL[4,5]*c5[i,j]
             -pem[i,j]*(AL[4,1]*c1[i,j]+AL[4,2]*c2[i,j]+AL[4,3]*c3[i,j]+AL[4,4]*c4[i,j]+AL[4,5]*c5
             -pem[i,j]*alpha*krate*c4[i,j]^2 ;
#
# Additional algebraic equations resulting from discretizing the boundary conditions
#
subject to AEc1{i in FE, j in CP} :
AL[1,1]*c1[i,j]+AL[1,2]*c2[i,j]+AL[1,3]*c3[i,j]+AL[1,4]*c4[i,j]+AL[1,5]*c5[i,j] - pem[i,j]*(c1[i,j
```

```
subject to AEc5{i in FE, j in CP} :
AL[5,1]*c1[i,j]+AL[5,2]*c2[i,j]+AL[5,3]*c3[i,j]+AL[5,4]*c4[i,j]+AL[5,5]*c5[i,j] = 0 ;
#
# Initial conditions constraints
#
subject to IVc2: c02[1] = c2init ;
subject to IVc3: c03[1] = c3init ;
subject to IVc4: c04[1] = c4init ;
#
# Constraint on the total transition time
#
subject to TTT: TIME = TransTime ;

# -- End of the pde.mod file --

#
# This file contains all the information to run one
# of the cases of a tubular reactor with diffusive and
# convective mass transfer dynamic optimization problem
#
#
# First order derivatives collocation matrix
#
param  A:        1                   2                   3 :=
      1  0.19681547722366   0.39442431473909   0.37640306270047
      2 -0.06553542585020   0.29207341166523   0.51248582618842
      3  0.02377097434822  -0.04154875212600   0.11111111111111;

param AL:     1                        2                        3                        4
```

```
1  -1.299999999999999e+001  1.478830557701236e+001 -2.666666666666668e+000  1.878361089654309e+000
2  -5.323790007724444e+000  3.872983346207410e+000  2.065591117977290e+000 -1.290994448735808e+000
3   1.499999999999999e+000 -3.227486121839514e+000  2.127927463864883e-015  3.227486121839517e+000
4  -6.762099922755483e-001  1.290994448735803e+000 -2.065591117977285e+000 -3.872983346207437e+000
5   9.999999999999968e-001 -1.878361089654300e+000  2.666666666666660e+000 -1.478830557701238e+001

param BL:     1                       2                       3                       4
1   8.399999999999994e+001 -1.220631667954075e+002  5.866666666666666e+001 -4.460349987125922e+001
2   5.323790007724445e+001 -7.333333333333327e+001  2.666666666666665e+001 -1.333333333333333e+001
3  -5.999999999999990e+000  1.666666666666666e+001 -2.133333333333333e+001  1.666666666666668e+001
4   6.762099922755510e+000 -1.333333333333336e+001  2.666666666666669e+001 -7.333333333333350e+001
5   2.399999999999996e+001 -4.460349987125911e+001  5.866666666666662e+001 -1.220631667954076e+002



let NFE       := 20  ;
let NCP       := 3   ;
let NPOC      := 5   ;
let TransTime := 0.1 ;

let r1 := 0.15505102572168;
let r2 := 0.64494897427832;
let r3 :=  1               ;
#
# Initial value fixed conditions and final (desired) conditions
#
let c2init := 1    ;
let c3init := 1    ;
let c4init := 1    ;
```

```
let peminit:= 2    ;
let c5des  := 0.13 ;
let pemdes := 96   ;
#
# Tubular reactor parameters
#
let alpha    := 1    ;
let krate    := 3.36 ;
let alphac5  := 1    ;
let alphapem := 1    ;
#
# Initial guesses of the decision variables
#
let {i in FE, j in CP} c1 [i,j]    := 1     ;
let {i in FE, j in CP} c2 [i,j]    := 1     ;
let {i in FE, j in CP} c3 [i,j]    := 1     ;
let {i in FE, j in CP} c4 [i,j]    := 1     ;
let {i in FE, j in CP} c5 [i,j]    := 1     ;
let {i in FE, j in CP} pem[i,j]    := 10    ;
let {i in FE}          c02[i]      := 1     ;
let {i in FE}          c03[i]      := 1     ;
let {i in FE}          c04[i]      := 1     ;
let {i in FE, j in CP} c2dot [i,j] := 1     ;
let {i in FE, j in CP} c3dot [i,j] := 1     ;
let {i in FE, j in CP} c4dot [i,j] := 1     ;
let {i in FE}          H[i]        := 1/NFE ;

#-- End of the pde.dat file --
```

Extension to Handle Grade Transitions in Polymerization Reactors

- Objective Function

$$\max \quad \left\{ \sum_{i=1}^{N_p} \frac{C_i^p W_i}{T_c} - \sum_{i=1}^{N_p} \frac{C_i^s (G_i - W_i/T_c)}{2\Theta_i} - \sum_{k=1}^{N_s} \sum_{f=1}^{N_{fe}} h_{fk} \sum_{c=1}^{N_{cp}} \frac{C^r t_{fck} \Omega_{c,N_{cp}}}{T_c} \left((x_{fck}^1 - \bar{x}_k^1)^2 \right.\right.$$
$$\left.\left. + \ldots + (x_{fck}^n - \bar{x}_k^n)^2 + (u_{fck}^1 - \bar{u}_k^1)^2 + \ldots + (u_{fck}^m - \bar{u}_k^m)^2 \right) \right\} \tag{7}$$

Transition Cost: $\frac{1}{T_c} \int_0^{t_f} \left[\sum_n (x^n - \bar{x}^n)^2 + \sum_m (u^m - \bar{u}^m)^2 \right] C^r dt$

discretized by Radau Quadrature as:

$$\sum_{k=1}^{N_s} \sum_{f=1}^{N_{fe}} h_{fk} \sum_{c=1}^{N_{pc}} \frac{C^r t_{fck} \Omega_{c,N_{cp}}}{T_c} \left((x_{fck}^1 - \bar{x}_k^1)^2 + \ldots + (x_{fck}^n - \bar{x}_k^n)^2 + (u_{fck}^1 - \bar{u}_k^1)^2 + \ldots + (u_{fck}^m - \bar{u}_k^m)^2 \right)$$

Scheduling Optimization Formulation

- Product Assignment

$$\sum_{k=1}^{N_S} y_{ik} = 1, \ \forall i \tag{8}$$

$$\sum_{i=1}^{N_p} y_{ik} = 1, \ \forall k \tag{9}$$

$$y'_{ik} = y_{i,k-1}, \ \forall i, k \neq 1 \tag{10}$$

$$y'_{i,1} = y_{i,N_S}, \ \forall i \tag{11}$$

- Amounts Manufactured

$$W_i \geqslant D_i T_c, \ \forall i \tag{12}$$

$$W_i = G_i \Theta_i, \ \forall i \tag{13}$$

$$G_i = F^o(1 - X_i), \ \forall i \tag{14}$$

Scheduling Optimization Formulation

- Processing Times

$$\theta_{ik} \leqslant \theta^{max} y_{ik}, \ \forall i, k \tag{15}$$

$$\Theta_i = \sum_{k=1}^{N_S} \theta_{ik}, \ \forall i \tag{16}$$

$$p_k = \sum_{i=1}^{N_p} \theta_{ik}, \ \forall k \tag{17}$$

- Transition between Products

$$z_{ipk} \geqslant y'_{pk} + y_{ik} - 1, \ \forall i, p, k \tag{18}$$

- Timing Relations

$$\theta_k^t = \sum_{i=1}^{N_p} \sum_{p=1}^{N_p} t_{pi}^t z_{ipk}, \ \forall k \tag{19}$$

$$t_1^s = 0 \tag{20}$$

$$t_k^e = t_k^s + p_k + \sum_{i=1}^{N_p} \sum_{p=1}^{N_p} t_{pi}^t z_{ipk}, \ \forall k \tag{21}$$

$$t_k^s = t_{k-1}^e, \ \forall k \neq 1 \tag{22}$$

$$t_k^e \leqslant T_c, \ \forall k \tag{23}$$

$$t_{fck} = (f-1)\frac{\theta_k^t}{N_{fe}} + \frac{\theta_k^t}{N_{fe}}\gamma_c, \ \forall f, c, k \tag{24}$$

Optimal Control Formulation

- Dynamic Mathematical Model Discretization

$$x^n_{fck} = x^n_{o,fk} + \theta^t_k h_{fk} \sum_{l=1}^{N_{cp}} \Omega_{lc} \dot{x}^n_{flk}, \ \forall n, f, c, k \tag{25}$$

- Continuity Constraint between Finite Elements

$$x^n_{o,fk} = x^n_{o,f-1,k} + \theta^t_k h_{f-1,k} \sum_{l=1}^{N_{cp}} \Omega_{l,N_{cp}} \dot{x}^n_{f-1,l,k}, \ \forall n, f \geqslant 2, k \tag{26}$$

- Model Behavior at each Collocation Point

$$\dot{x}^n_{fck} = f^n(x^1_{fck}, \ldots, x^n_{fck}, u^1_{fck}, \ldots u^m_{fck}), \ \forall n, f, c, k \tag{27}$$

Optimal Control Formulation

- Initial/Final Controlled/Manipulated Variables at Each Slot

$$x^n_{in,1} = \sum_{i=1}^{N_p} x^n_{ss,i} y_{i,N_s}, \ \forall n \tag{28}$$

$$x^n_{in,k} = \sum_{i=1}^{N_p} x^n_{ss,i} y_{i,k-1}, \ \forall n, k \neq 1 \tag{29}$$

$$\bar{x}^n_k = \sum_{i=1}^{N_p} x^n_{ss,i} y_{i,k}, \ \forall n, k \tag{30}$$

$$u^m_{in,1} = \sum_{i=1}^{N_p} u^m_{ss,i} y_{i,N_s}, \ \forall m \tag{31}$$

$$u^m_{in,k} = \sum_{i=1}^{N_p} u^m_{ss,i} y_{i,k-1}, \ \forall m, k \neq 1 \tag{32}$$

$$\bar{u}^m_k = \sum_{i=1}^{N_p} u^m_{ss,i} y_{i,k}, \ \forall m, k \tag{33}$$

$$u^m_{1,1,k} = u^m_{in,k}, \ \forall m,k \tag{34}$$

$$u^m_{N_{fe},N_{cp},k} = \bar{u}^m_{in,k}, \ \forall m,k \tag{35}$$

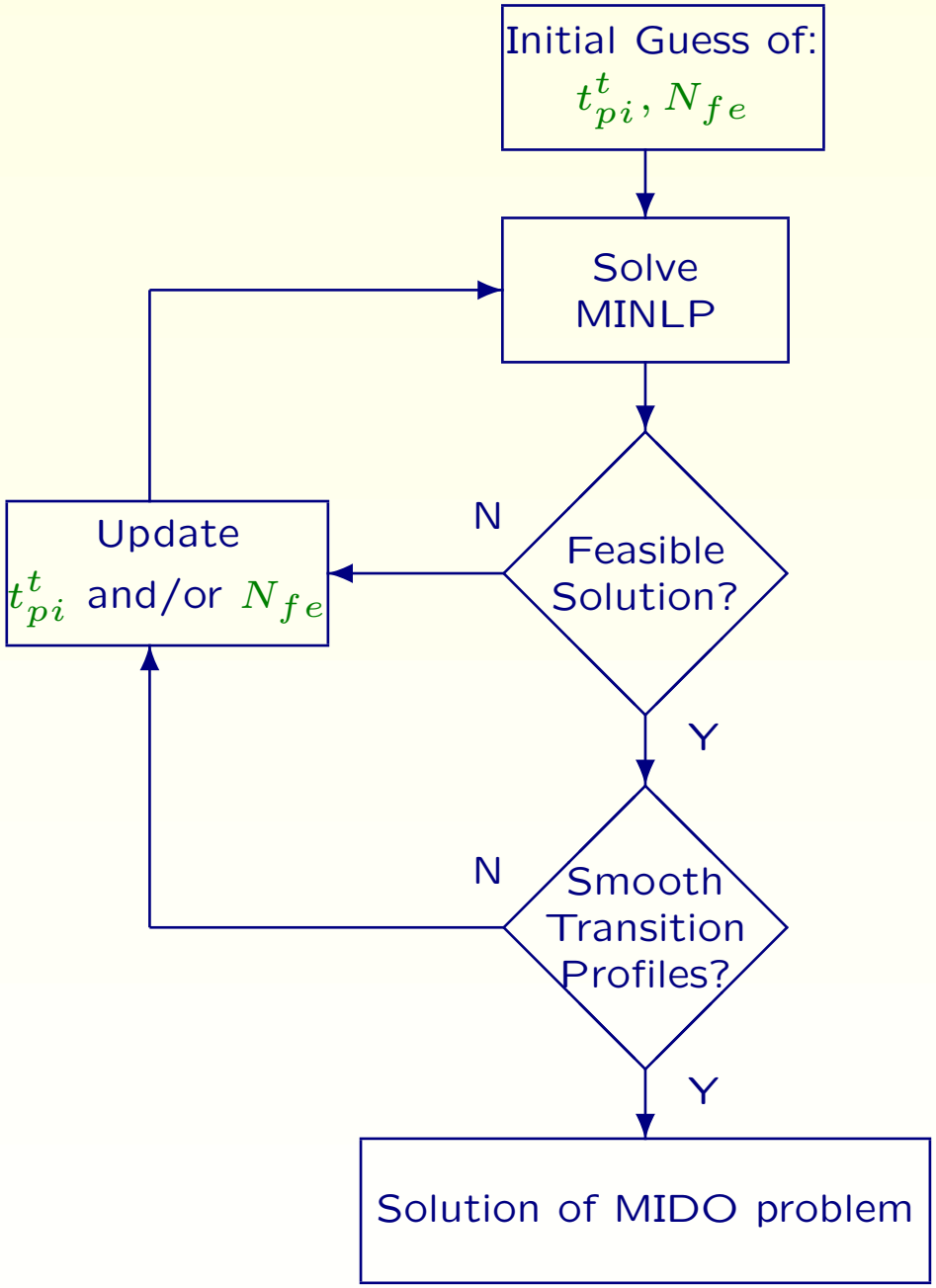
$$x^n_{o,1,k} = x^n_{in,k}, \ \forall n,k \tag{36}$$

- Upper and Lower Bounds on the Decision Variables

$$x^n_{min} \leqslant x^n_{fck} \leqslant x^n_{max}, \forall n,f,c,k \tag{37}$$

$$u^m_{min} \leqslant u^m_{fck} \leqslant u^m_{max}, \forall m,f,c,k \tag{38}$$

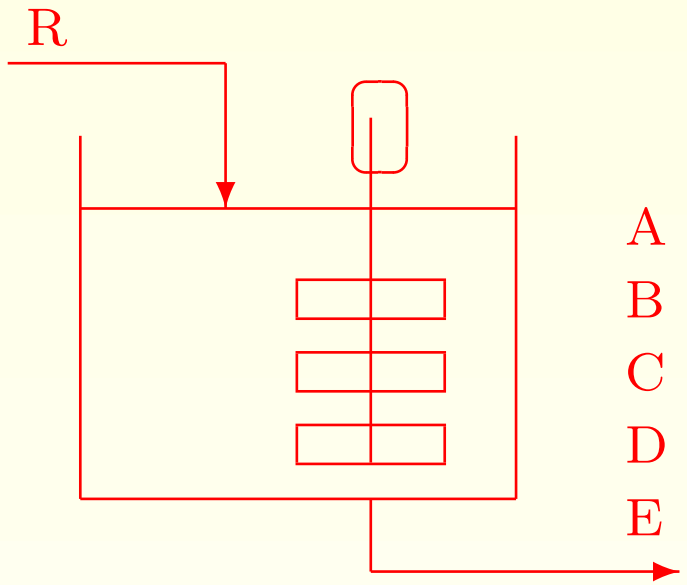
Solution Algorithm



Example: CSTR with a Simple Irreversible Reaction

$$3R \xrightarrow{k} P, \quad -\mathcal{R}_R = kC_R^3$$

$$\frac{dC_R}{dt} = \frac{Q}{V}(C_o - C_R) + \mathcal{R}_R$$



Process data

Product	Q [lt/hr]	C_R [mol/lt]	Demand rate [Kg/h]	Product cost [$/kg]	Inventory cost [$]
A	10	0.0967	3	200	1
B	100	0.2	8	150	1.5
C	400	0.3032	10	130	1.8
D	1000	0.393	10	125	2
E	2500	0.5	10	120	1.7

Results

Best Solution: $A \rightarrow E \rightarrow D \rightarrow C \rightarrow B$, Profit= $ 7889, Cyclic time=124.8 h

Slot	Product	Process time [h]	Production rate [Kg/h]	w [Kg]	Transition Time [h]	T start [h]	T end [h]
1	A	41.5	9.033	374.31	5	0	46.4
2	E	23.3	1250	29162.3	5	46.4	74.7
3	D	2.06	607	1247.7	5	74.7	81.8
4	C	4.48	278.72	1247.7	5	81.8	91.2
5	B	12.48	80	998.2	21	91.2	124.7

Second Best Solution: $A \rightarrow D \rightarrow E \rightarrow C \rightarrow B$, Profit= $ 7791, Cycle time= 125 h

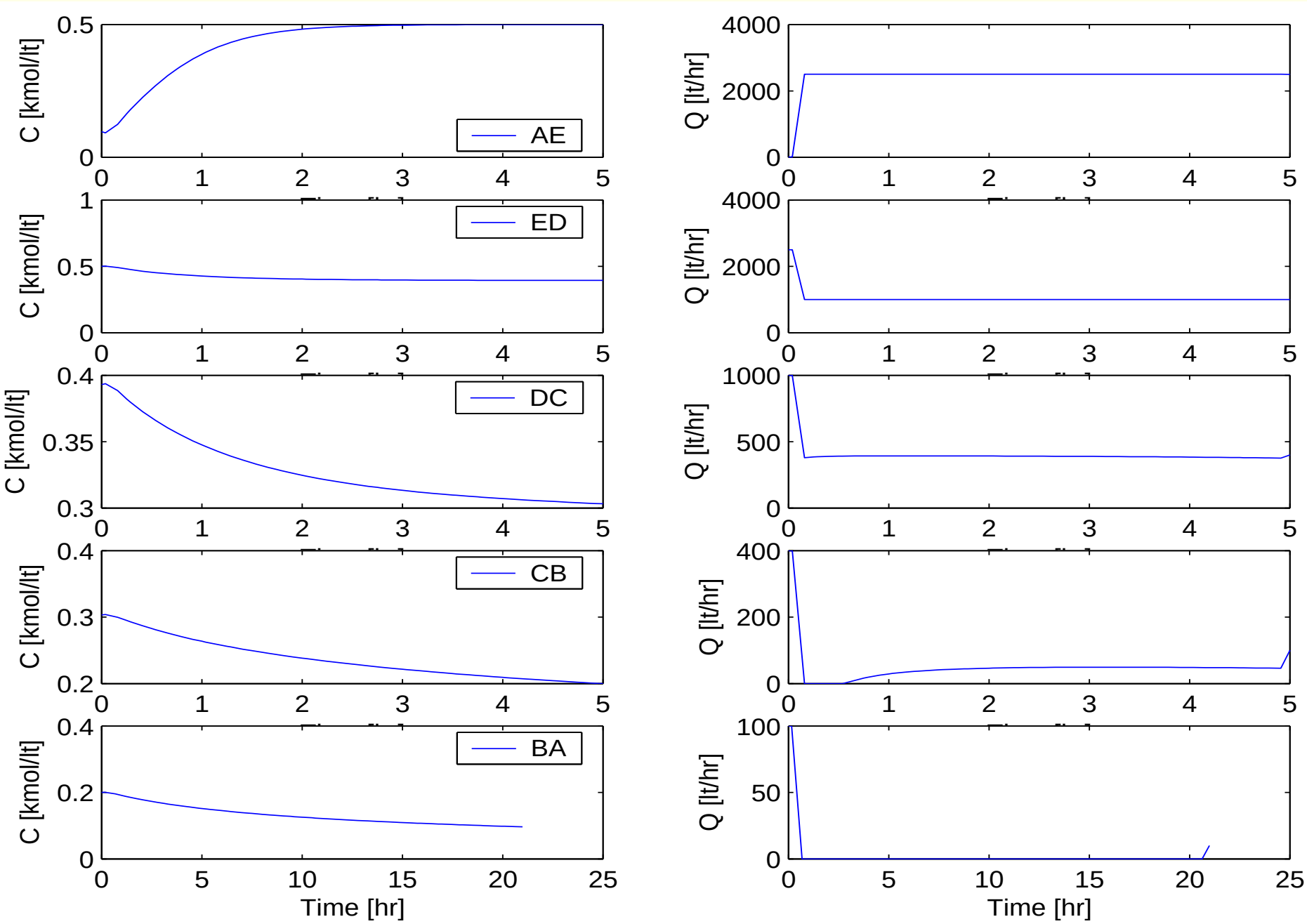
Slot	Product	Process time [h]	Production rate [Kg/h]	w [Kg]	Transition Time [h]	T start [h]	T end [h]
1	A	41.5	9.033	374.31	5	0	46.4
2	D	2.06	607	1249.4	5	46.4	53.6
3	E	23.4	1250	29270.4	5	53.6	82
4	C	4.48	278.72	1249.4	5	82	91.5
5	B	12.48	80	999.5	21	91.5	125

Results

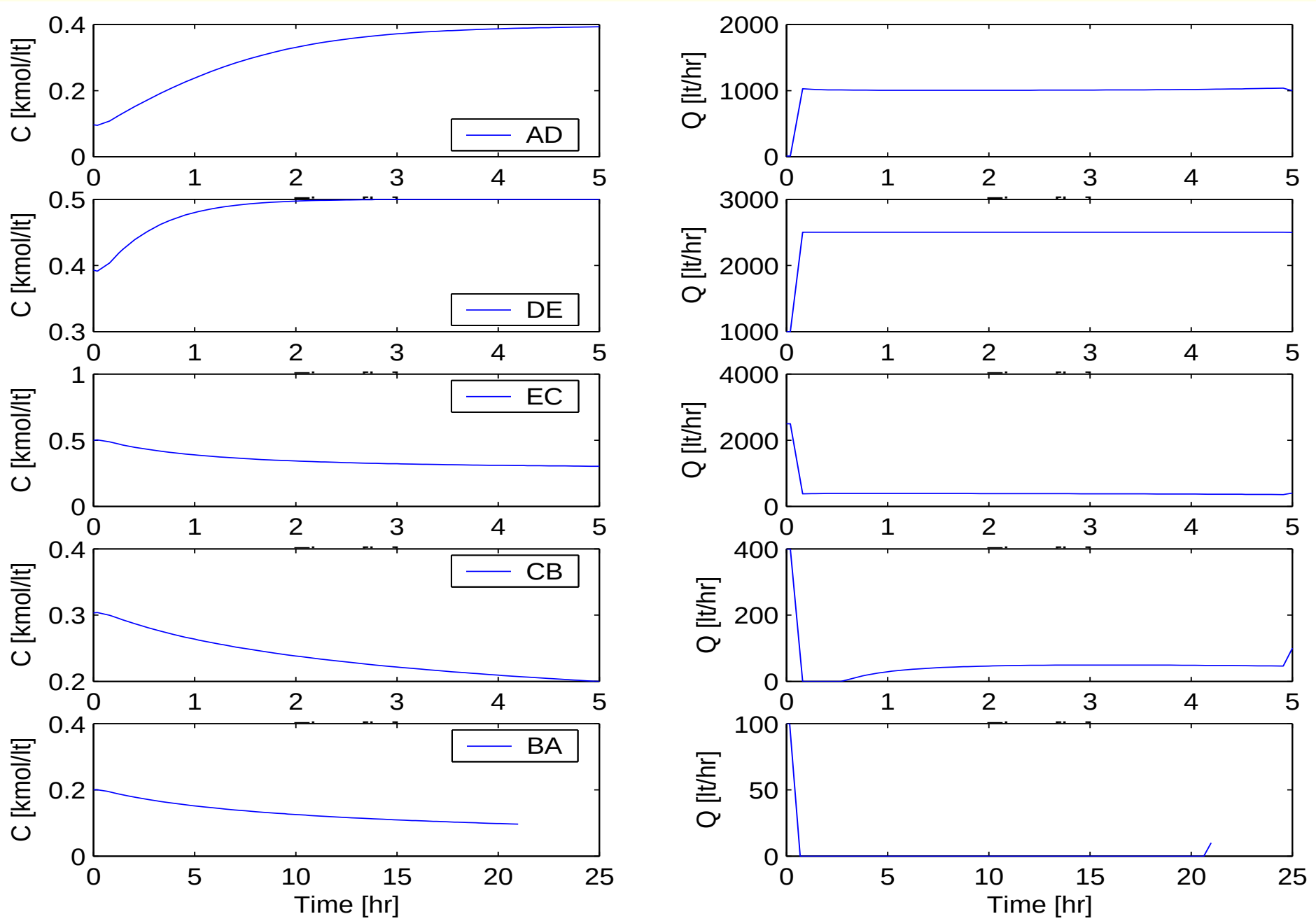
Third Best Solution: $B \rightarrow A \rightarrow E \rightarrow C \rightarrow D$, Profit= $ 6821.6, Cycle time= 127 h

Slot	Product	Process time [h]	Production rate [Kg/h]	w [Kg]	Transition Time [h]	T start [h]	T end [h]
1	B	12.7	80	1012.5	21	0	33.7
2	A	42.04	9.033	379.7	5	33.7	80.7
3	E	23.3	1250	29125.4	5	80.7	109
4	C	4.6	278.72	1265.6	5	109	118.6
5	D	2.09	607	1265.6	6	118.6	127

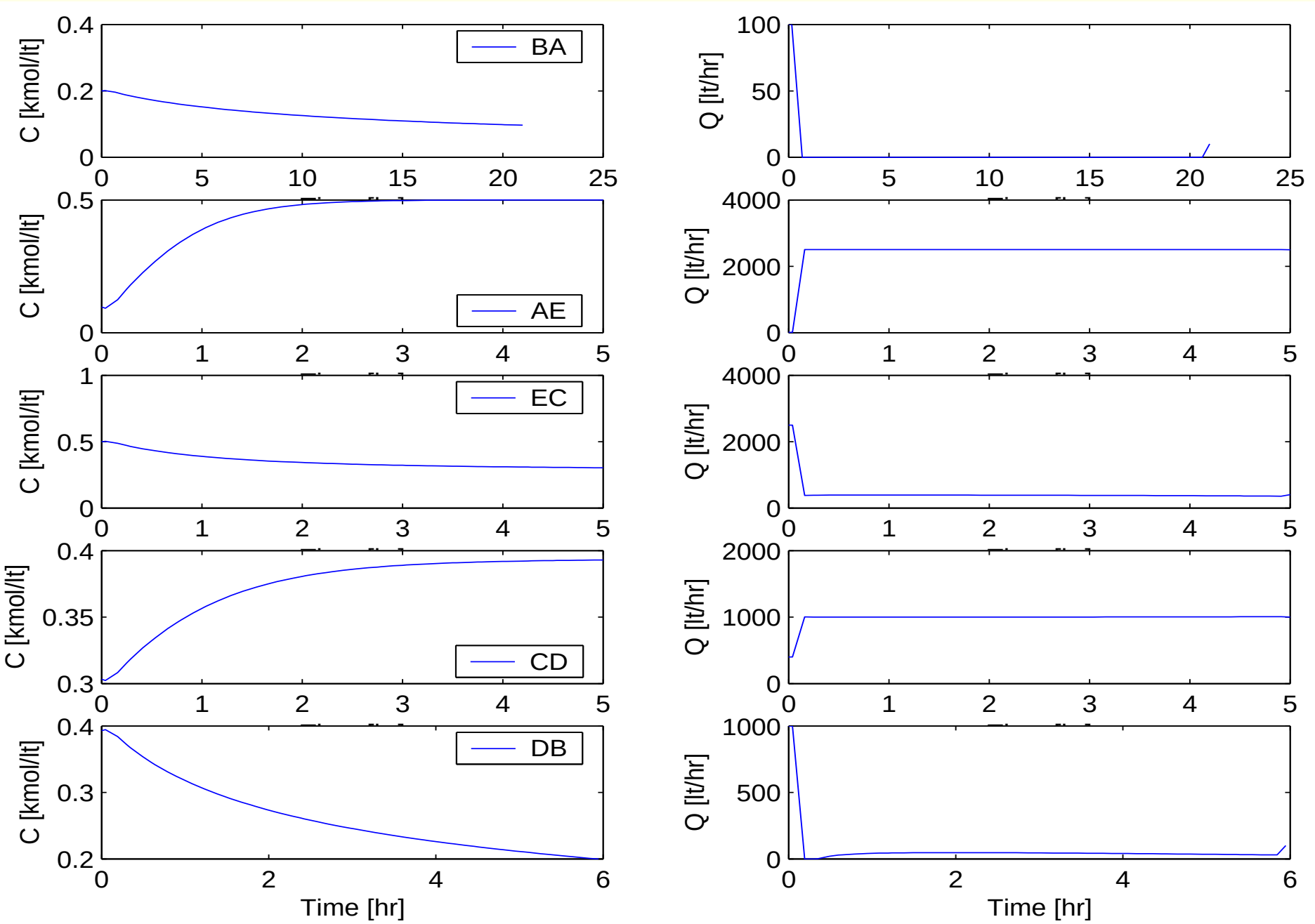
Optimal transition profiles first solution



Optimal transition profiles second solution



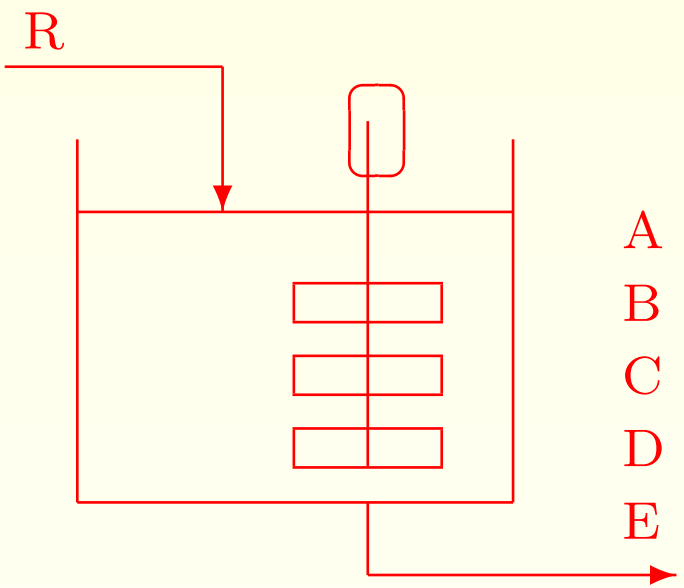
Optimal transition profiles third solution

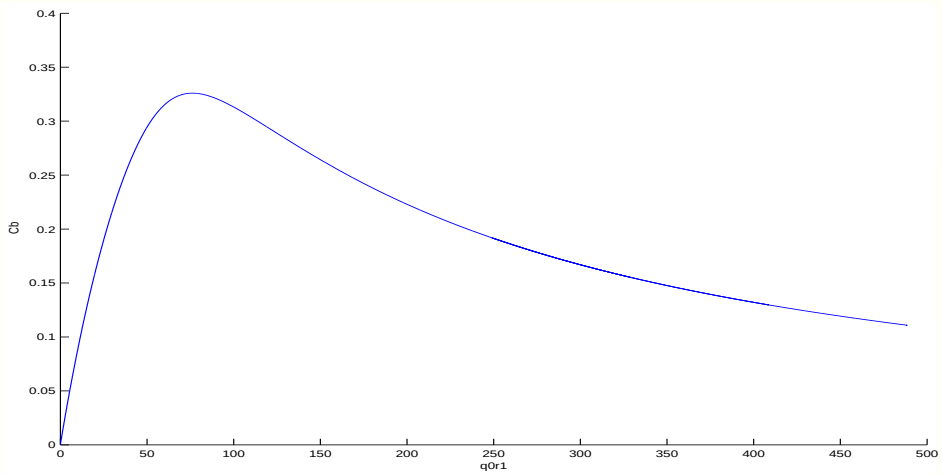


Example: CSTR with simultaneous reactions and input multiplicities

$$
\begin{aligned}
2R_1 &\xrightarrow{k_1} A \\
R_1 + R_2 &\xrightarrow{k_2} B \\
R_1 + R_3 &\xrightarrow{k_3} C
\end{aligned}
$$

$$
\begin{aligned}
\frac{dC_{R_1}}{dt} &= \frac{(Q_{R_1} C^i_{R_1} - Q C_{R_1})}{V} + \mathcal{R}_{r_1} \\
\frac{dC_{R_2}}{dt} &= \frac{(Q_{R_2} C^i_{R_2} - Q C_{R_2})}{V} + \mathcal{R}_{r_2} \\
\frac{dC_{R_3}}{dt} &= \frac{(Q_{R_3} C^i_{R_3} - Q C_{R_3})}{V} + \mathcal{R}_{r_3} \\
\frac{dC_A}{dt} &= \frac{Q(C^i_A - C_A)}{V} + \mathcal{R}_A \\
\frac{dC_B}{dt} &= \frac{Q(C^i_B - C_B)}{V} + \mathcal{R}_B \\
\frac{dC_C}{dt} &= \frac{Q(C^i_C - C_C)}{V} + \mathcal{R}_C
\end{aligned}
$$





Example: CSTR with simultaneous reactions and input multiplicities

Process data

Prod	Q_{R_1}	Q_{R_2}	Q_{R_3}	C_{R_1}	C_{R_2}	C_{R_3}	C_A	C_B	C_C
A	100	0	0	0.333	0	0	0.666	0	0
B	100	100	0	0.1335	0.0869	0	0.0534	0.3131	0
C	100	0	100	0.0837	0	0.1048	0.021	0	0.3951

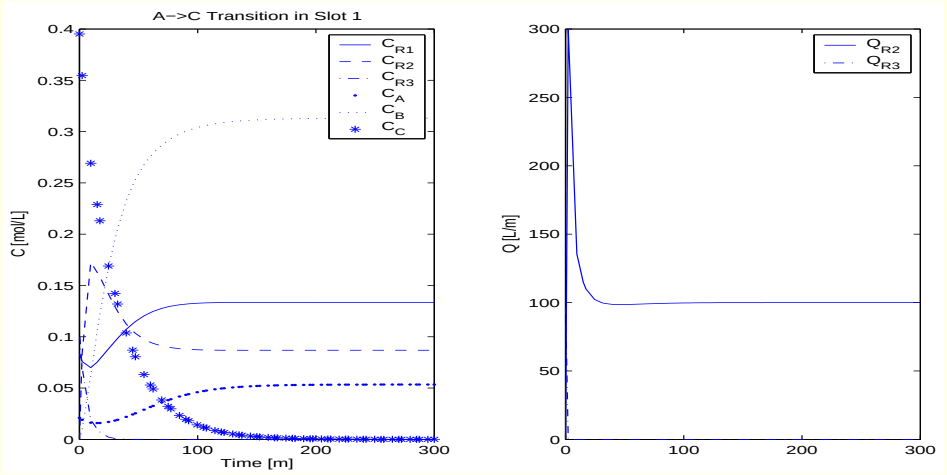
Demand rate and cost information

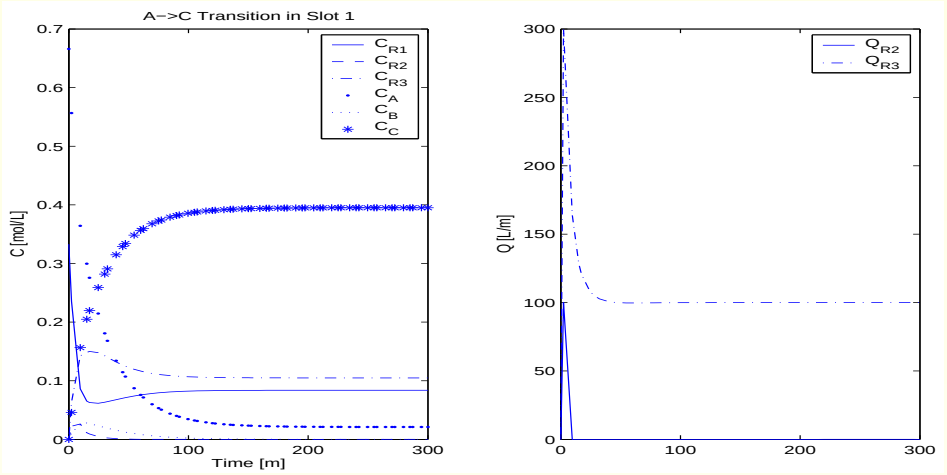
Product	Demand [Kg/m]	Product cost [$/kg]	Inventory cost [$]
A	5	500	1
B	10	400	1.5
C	15	600	1.8

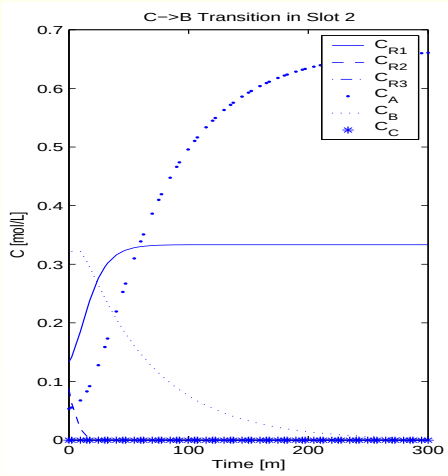
Profit= $ 32388, Cyclic time= 317.5 m

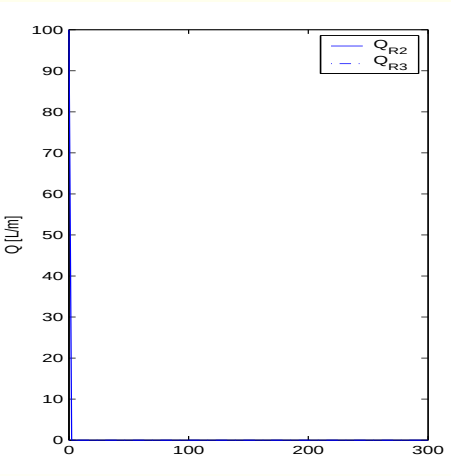
Slot	Product	Process time [m]	Production rate [Kg/m]	w Time [Kg]	Transition [m]	T start [m]	T end [m]
1	C	204.2	89.52	18273.3	15	0	219.2
2	B	44.5	71.31	3174.4	15	219.2	278.7
3	A	23.8	66.7	1587.2	15	278.7	317.5

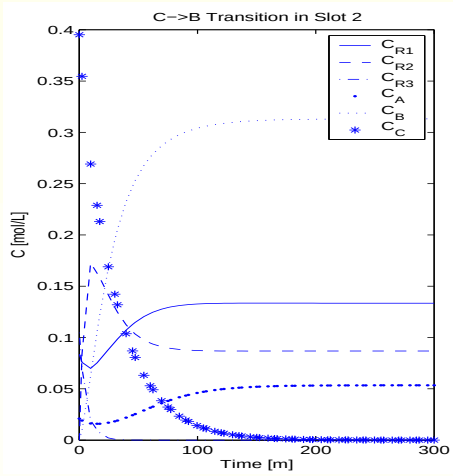
Example: CSTR with simultaneous reactions and input multiplicities

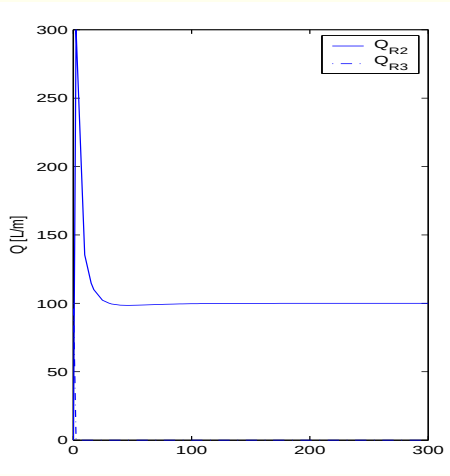


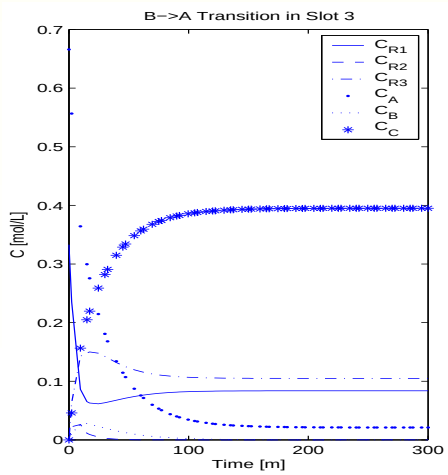


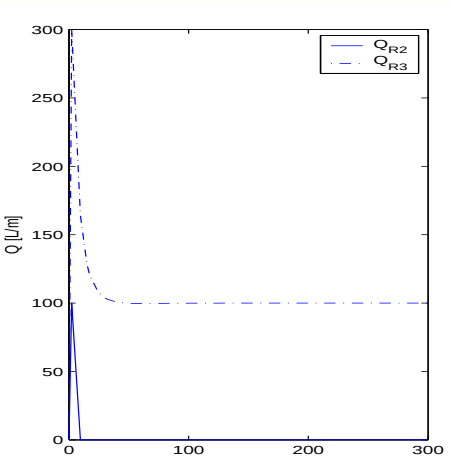


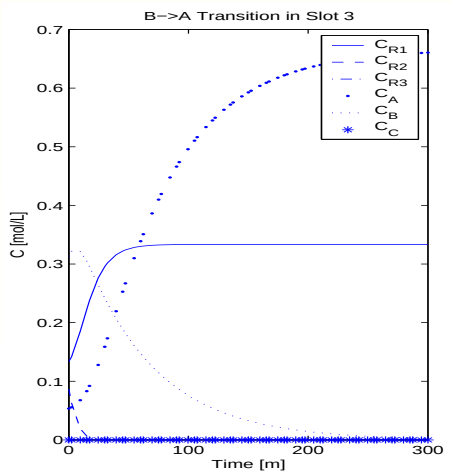


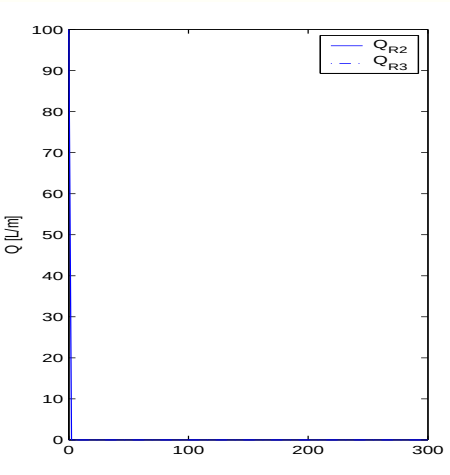








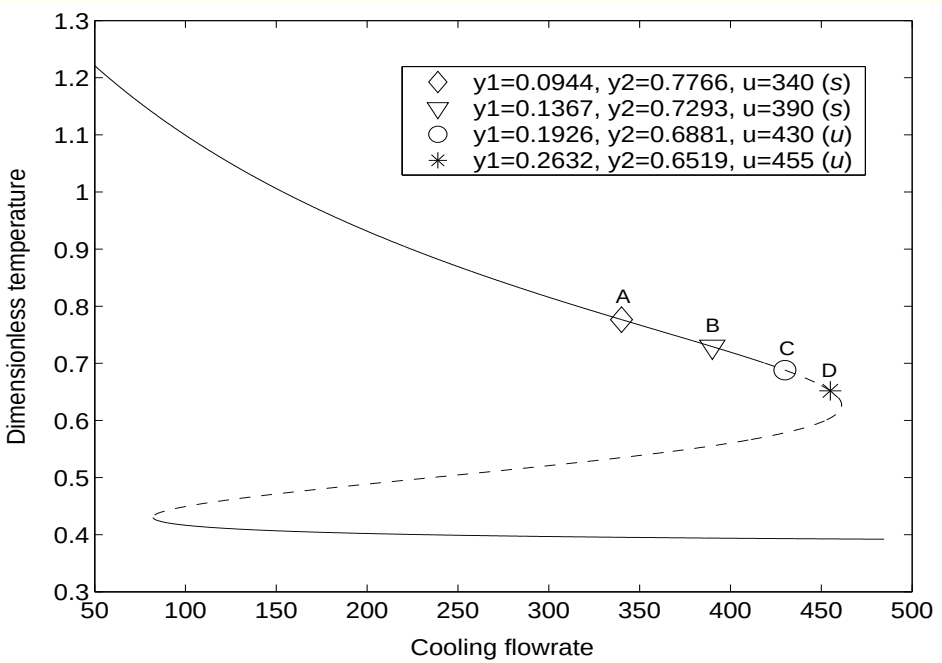




Example: CSTR with output multiplicities

$$\frac{dy_1}{dt} = \frac{1 - y_1}{\theta} - k_{10} e^{-N/y_2} y_1$$

$$\frac{dy_2}{dt} = \frac{y_f - y_2}{\theta} + k_{10} e^{-N/y_2} y_1 - \alpha u (y_2 - y_c)$$



Parameter values

θ	20	Residence time	T_f	300	Feed temperature
J	100	$(-\Delta H)/(\rho C_p)$	k_{10}	300	Preexponential factor
c_f	7.6	Feed concentration	T_c	290	Coolant temperature
α	1.95x10^{-4}	Dimensionless heat transfer area	N	5	$E_1/(RJc_f)$

Example: CSTR with output multiplicities

Process data

Product	Demand [Kg/h]	Product cost [$/kg]	Inventory cost [$]
A	100	100	1
B	200	50	1.3
C	400	30	1.4
D	500	80	1.1

Best Solution: Profit= $7657, Cyclic time= 100.6 h

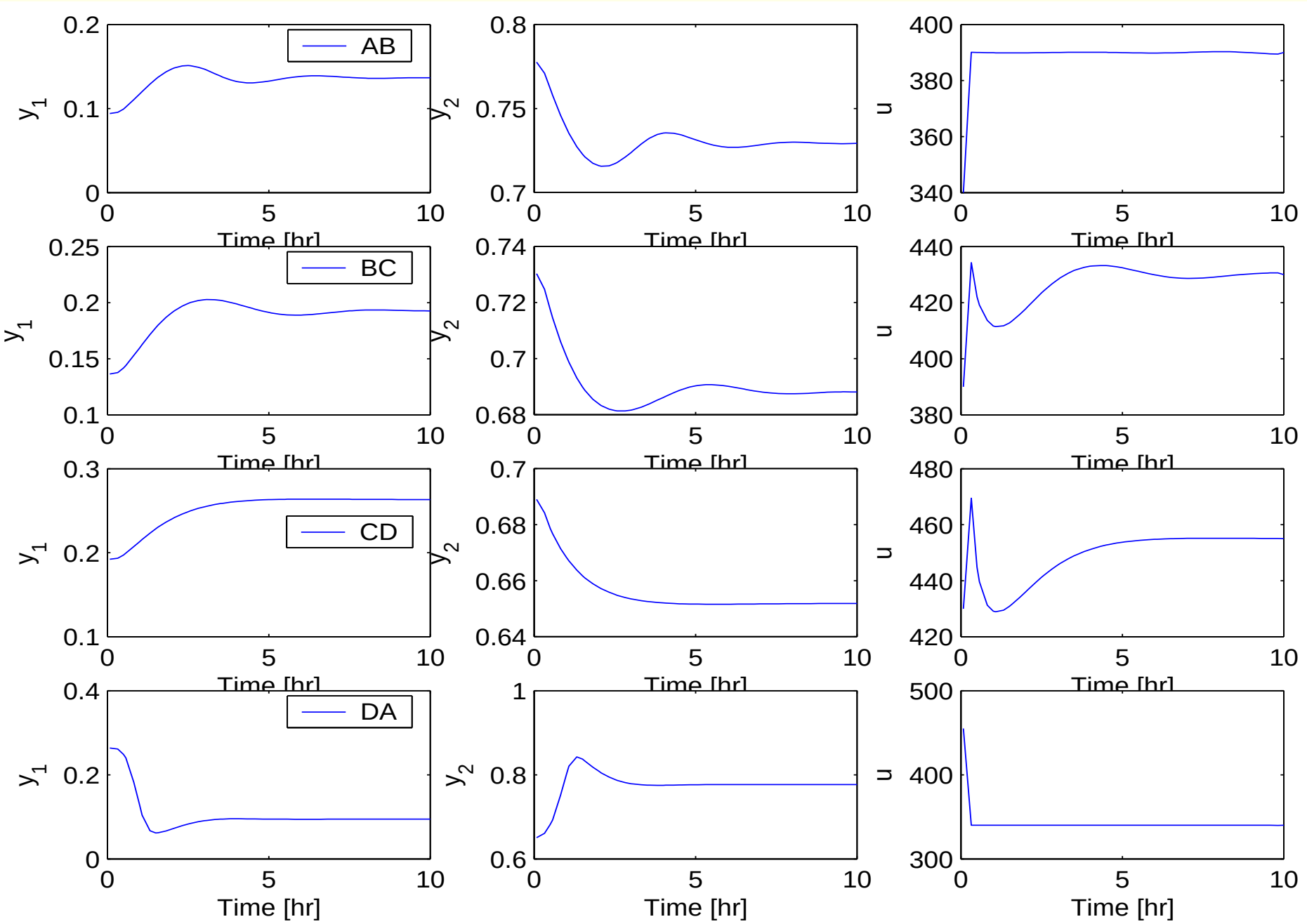
Slot	Product	Process time [h]	Production rate [Kg/h]	w Time [Kg]	Transition [h]	T start [h]	T end [h]
1	A	28.3	559.9	15831.7	10	0	38.3
2	B	13.1	613.6	8044.9	10	38.3	61.4
3	C	13.4	656.1	8748.9	10	61.4	84.8
4	D	5.8	688.3	4022.5	10	84.8	100.6

Example: CSTR with output multiplicities

Second Best Solution: Profit= \$6070.6, Cyclic time= 104.4 h

Slot	Product	Process time [h]	Production rate [Kg/h]	w [Kg]	Transition Time [h]	T start [h]	T end [h]
1	D	6.07	559.9	4176.7	10	0	16.07
2	A	28.9	613.6	16177.2	10	16.07	55
3	C	13.9	656.1	9084.3	12	55	80.8
4	B	13.7	688.3	8353.4	10	80.8	104.4

Example: CSTR with output multiplicities



Example: High Impact Polystyrene (HIPS)

$$
\begin{aligned}
\frac{dx_1}{dt} &= \frac{Q_{imax}x_{1o}u_i - Q_{max}x_1u_f}{V} - k_d x_1 \\
\frac{dx_2}{dt} &= Q_{max}u_f\frac{x_{2o}-x_2}{V} - k_p x_2\left(\mu^r_{max}x_6 + \mu^0_{bmax}x_7\right) \\
\frac{dx_3}{dt} &= Q_{max}u_f\frac{x_{3o}-x_3}{V} - x_3(k_{I2}C_{rmax}x_4 + k_{fs}\mu^0_{rmax}x_6 + k_{fb}\mu^0_{bmax}x_7) \\
\frac{dx_4}{dt} &= 2f_a\frac{k_d C_{imax}x_1}{C_{rmax}} - x_4(k_{I1}C_{mmax}x_2 + k_{I2}C_{bmax}x_3) \\
\frac{dx_5}{dt} &= \frac{C_{bmax}x_3}{C_{brmax}}(k_{I2}C_{rmax}x_4 + k_{fb}(\mu^0_{rmax}x_6 + \mu^0_{bmax}x_7)) - x_5(k_{I3}C_{mmax}x_2 + k_t \\
&\quad (\mu^0_{rmax}x_6 + \mu^0_{bmax}x_7 + C_{bmax}x_5)) \\
\frac{dx_6}{dt} &= \frac{2k_{I0}(C_{mmax}x_2)^2 + k_{I1}C_{rmax}x_4C_{mmax}x_2 + C_{mmax}x_2k_{fs}\mu^0_{rmax}x_6\mu^0_{bmax}x_7}{\mu^0_{rmax}} \\
&\quad - (k_pC_{mmax}x_2 + k_t(\mu^0_{rmax}x_6 + \mu^0_{bmax}x_7 + C_{brmax}x_5) + k_{fs}C_{mmax}x_2 \\
&\quad + k_{fb}C_{bmax}x_3)x_6 + k_pC_{mmax}x_2x_6 \\
\frac{dx_7}{dt} &= \frac{k_{I3}C_{brmax}C_{mmax}x_5x_2}{\mu^0_{bmax}} - (k_pC_{mmax}x_2 + k_t(\mu^0_{rmax}x_6 + \mu^0_{bmax}x_7 + C_{brmax}x_5) \\
&\quad + k_{fs}C_{mmax}x_2k_{fb}C_{bmax}x_3)x_7 + k_pC_{mmax}x_2x_7
\end{aligned}
$$

Example: High Impact Polystyrene (HIPS)

V	6000	Reactor volume [L]
Q_i	$1.5\text{x}10^{3}$	Initiator flow rate [L/s]
C_{m0}	8.63	Monomer feed stream concentration [mol/L]
C_{b0}	1.05	Butadiene feed stream concentration [mol/L]
C_{I0}	0.98	Initiator feed stream concentration [mol/L]
T	377.5	Reactor temperature [K]
k_d	$7.28\text{x}10^{-4}$	Initiation reaction constant [1/s]
k_{I0}	$1.59\text{x}10^{-11}$	Initiation reaction constant [L/mol-s]
k_{I1}	$8.04\text{x}10^{2}$	Initiation reaction constant[L/mol-s]
k_{I2}	$1.61\text{x}10^{2}$	Initiation reaction constant[L/mol-s]
k_{I3}	$8.04\text{x}10^{2}$	Initiation reaction constant[L/mol-s]
k_p	$8.04\text{x}10^{2}$	Propagation reaction constant [L/mol-s]
k_{fs}	$2.99\text{x}10^{-1}$	Monomer transfer reaction constant [L/mol-s]
C_{mmax}	7.31	Maximum value of monomer concentration [mol/l]
C_{Imax}	$3\text{x}10^{-4}$	Maximum value of initiator concentration [mol/l]
C_{bmax}	1.05	Maximum value of butadiene concentration [mol/l]
C_{rmax}	$6.29\text{x}10^{-11}$	Maximum value of radical concentration [mol/l]
C_{brmax}	$4.97\text{x}10^{-12}$	Maximum value of butadiene radical concentration [mol/l]
μ^0_{rmax}	$8.66\text{x}10^{-8}$	Maximum value of zero radical death moment
μ^0_{bmax}	$4.41\text{x}10^{-9}$	Maximum value of zero butadiene radical death moment
Q_{Imax}	$1.5\text{x}10^{-3}$	Maximum value of initiator flow rate [l/s]
Q_{mmax}	1.14	Maximum value of feed stream flow rate [l/s]

Example: High Impact Polystyrene (HIPS)

HIPS Grade Design Information

Grade	Q [l/s]	Conv.	Demand [kg/h]	Inv. Cost	Monomer Cost	Initiator Cost	Price
A	1.14	15	50	0.15	1	10	3.2
B	0.75	25	60	0.20	1	10	4.3
C	0.56	35	65	0.15	1	10	4.5
D	0.60	40	70	0.10	1	10	5.0
E	0.53	45	60	0.25	1	10	5.5

$E \rightarrow A \rightarrow B \rightarrow C \rightarrow D$, Profit= \$ 1456, Cyclic time=32.2 h

Product	Process T [h]	production [kg]	Trans T [h]	T start [h]	T end [h]
E	2.48	1937	1.34	0	3.83
A	2.87	1614	1.15	3.83	7.85
B	3.17	1937	1.11	7.85	12.14
C	3.10	2099	0.58	12.14	15.82
D	15.81	11370	0.67	15.82	32.29

Example: High Impact Polystyrene (HIPS)

Performance indicators for HIPS CSTR optimal and other suboptimal feasible solutions

Solution	sequence	Profit	Tc [h]	$\frac{Transtime}{CycTime}$	$\frac{w_D}{w_all}$	$\frac{w_{A,B,C,E}}{demand}$	$\frac{w_D}{demand}$
Optimal	E A B C D	1456	32.3	0.15	0.60	1.0	5.0
Sol.B	D A B C E	1352	33.0	0.16	0.59	1.0	4.9
Sol.C	E B A C D	1221	36.2	0.17	0.59	1.0	4.8
Sol.D	A B C D E	1155	37.2	0.18	0.59	1.0	4.7
Sol.E	E A C B D	1101	38.1	0.19	0.58	1.0	4.7
Sol.F	B A E C D	1045	39.0	0.20	0.58	1.0	4.6

Dominant eigenvalues for the base case (V=6000 L) and modified (V=2500 L)

Grade	Dominant Eigenvalue (Base case)	Dominant Eigenvalue (Modified case)
A	-1.59x10^{-4}	-3.92x10^{-4}
B	-1.02x10^{-4}	-2.55x10^{-4}
C	-7.97x10^{-5}	-2.10x10^{-4}
D	-7.30x10^{-5}	-1.88x10^{-4}
E	-6.96x10^{-5}	-1.78x10^{-4}

Example: High Impact Polystyrene (HIPS)

Profit= $ 1416, Cyclic timew=33 h

Product	Process T [h]	production [kg]	Trans T [h]	T start[h]	T end [h]
C	3.16	2141	0.58	0	3.75
D	16.03	11530	0.67	3.75	20.44
E	2.54	1977	1.54	20.44	24.52
A	2.93	1647	1.14	24.52	28.60
B	3.23	1977	1.11	28.60	32.95

Comparison between simultaneous and sequential solutions

Method	Sales [$/hr]	inv. costs [$/hr]	trans. costs [$/hr]	Profit [$/hr]
Simultaneous	2801.24	941.00	404.68	1455.55
Sequential	2790.51	959.99	414.19	1416.33

Example: High Impact Polystyrene (HIPS)

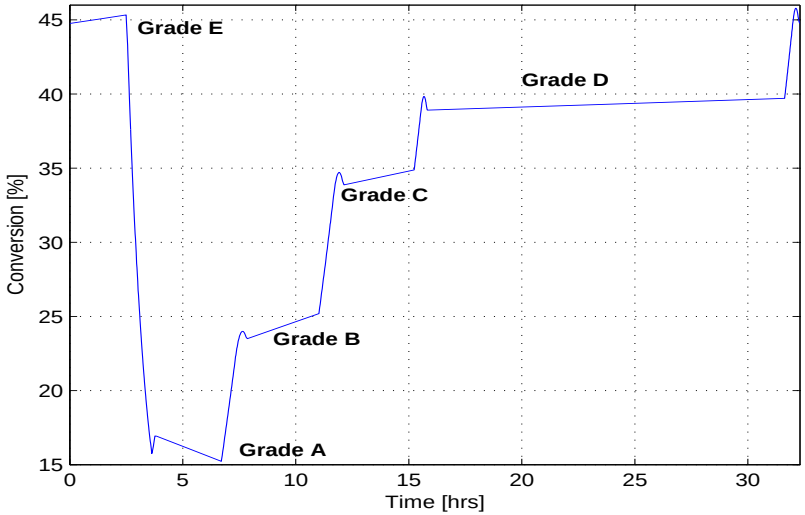
Comparison between simultaneous and modified sequential solutions

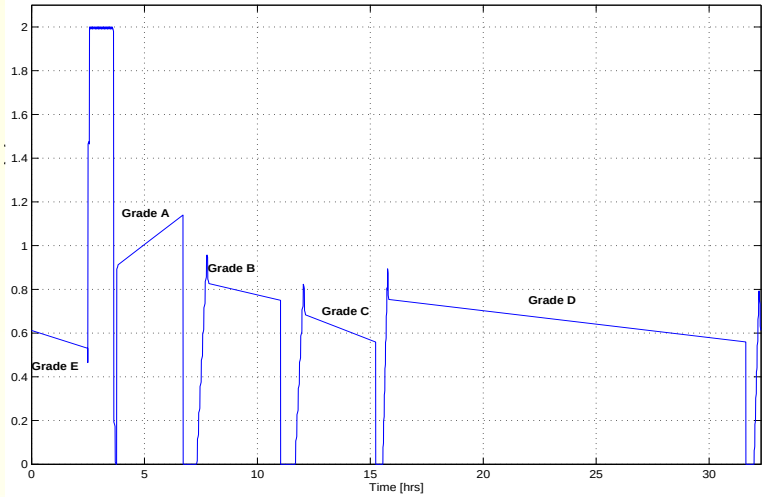
Method	Sales [$/hr]	inv. costs [$/hr]	trans. costs [$/hr]	Profit [$/hr]
Simultaneous	2801.24	941.00	404.68	1455.55
Sequential	2800.82	940.79	405.17	1454.86

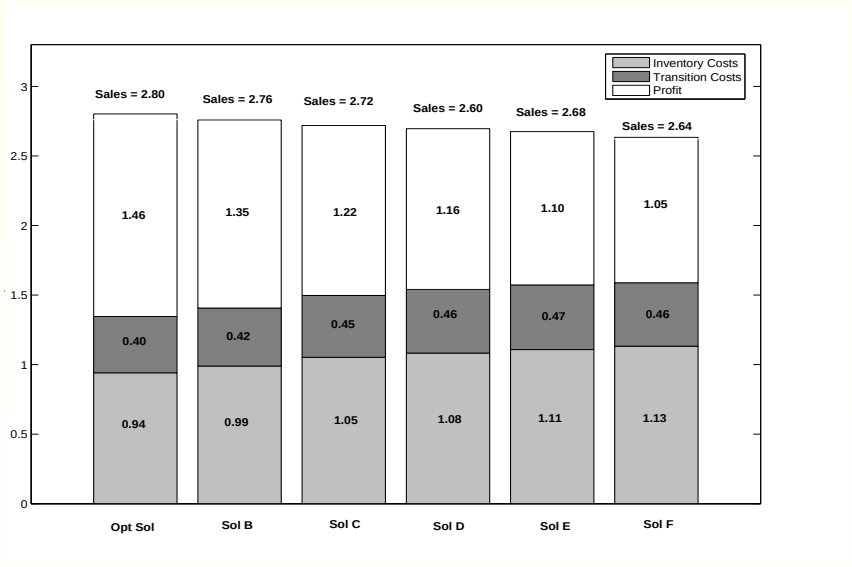
Solution using iterative approach, Profit = $ 906, Cyclic time= 40 h

Product	Process T [h]	production [kg]	Trans T [h]	T start[h]	T end [h]
A	3.57	2008	3	0	6.57
D	10.67	7686	3	6.57	20.26
C	3.86	2610	3	20.26	27.11
E	3.10	2409	3	27.11	33.21
B	3.92	2409	3	33.21	40.15

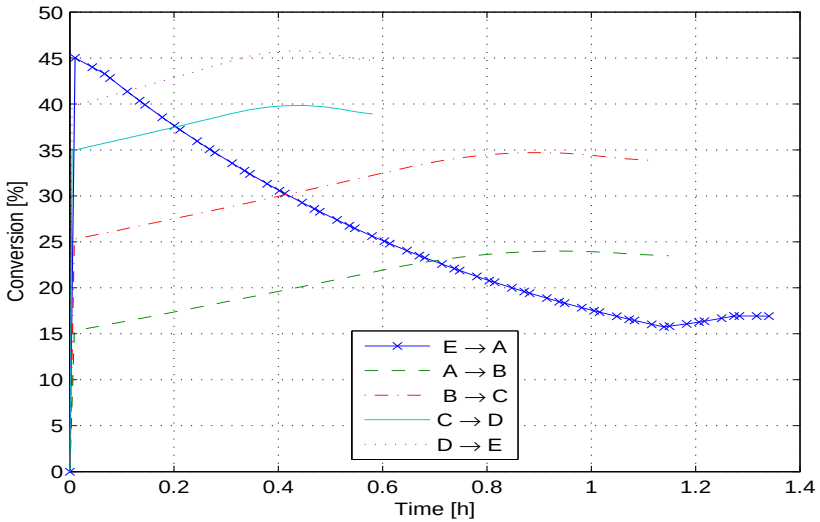
Example: High Impact Polystyrene (HIPS)

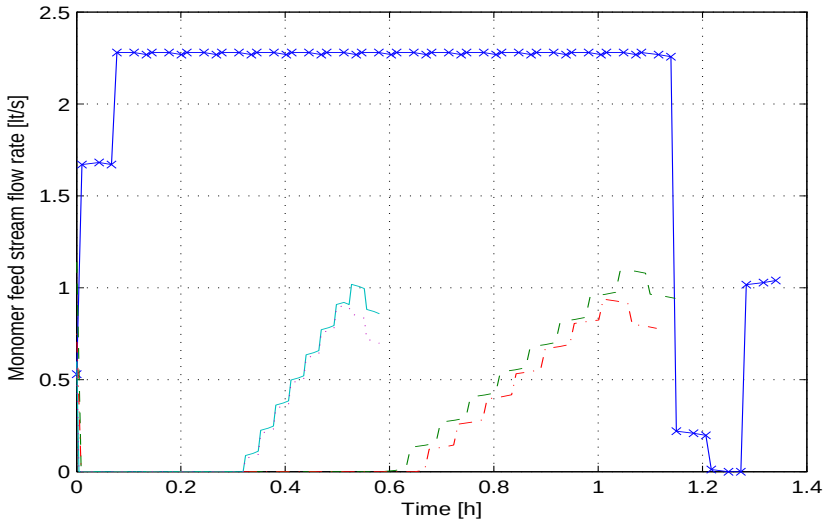


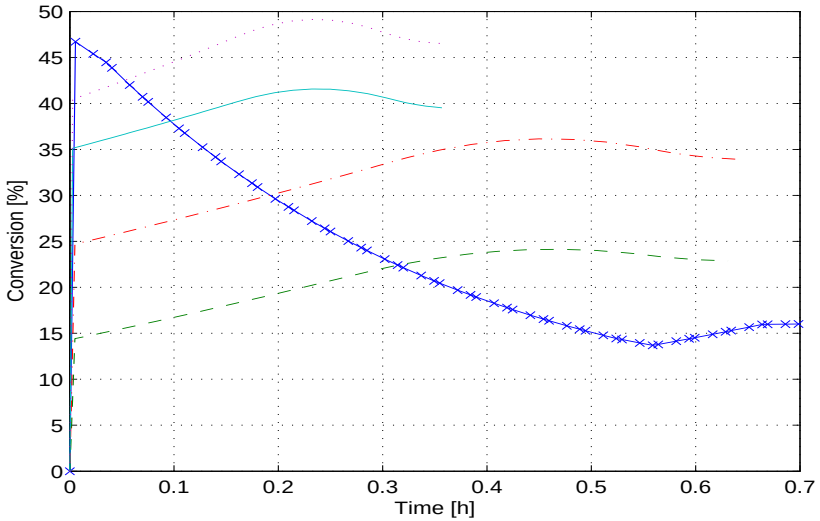


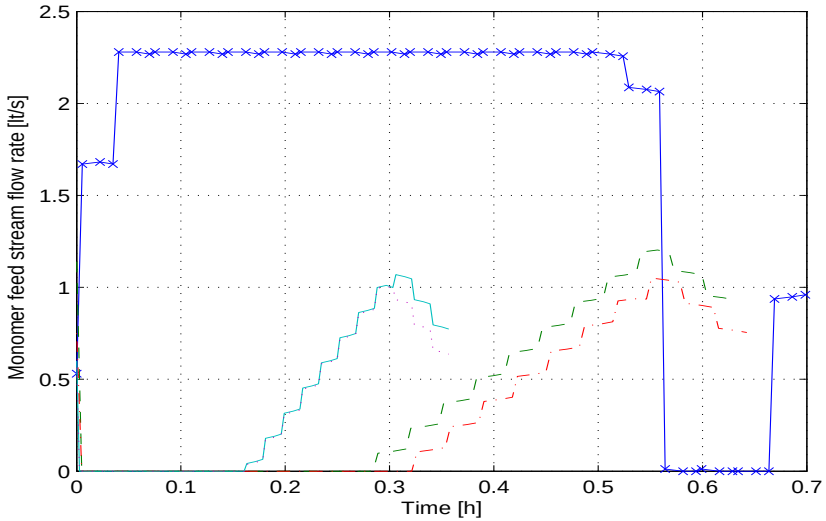


Example: High Impact Polystyrene (HIPS)

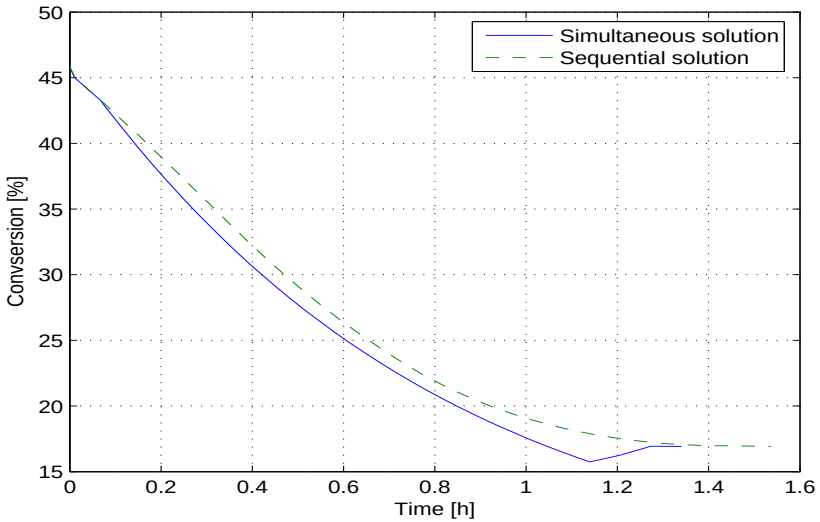


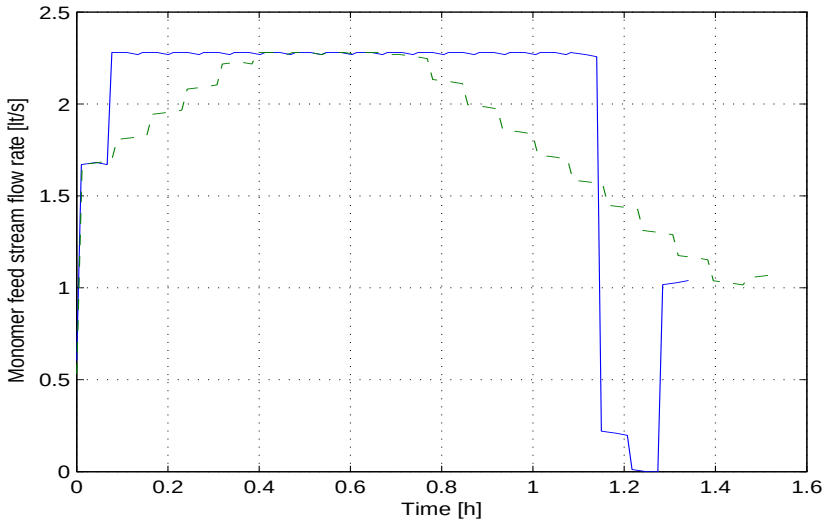


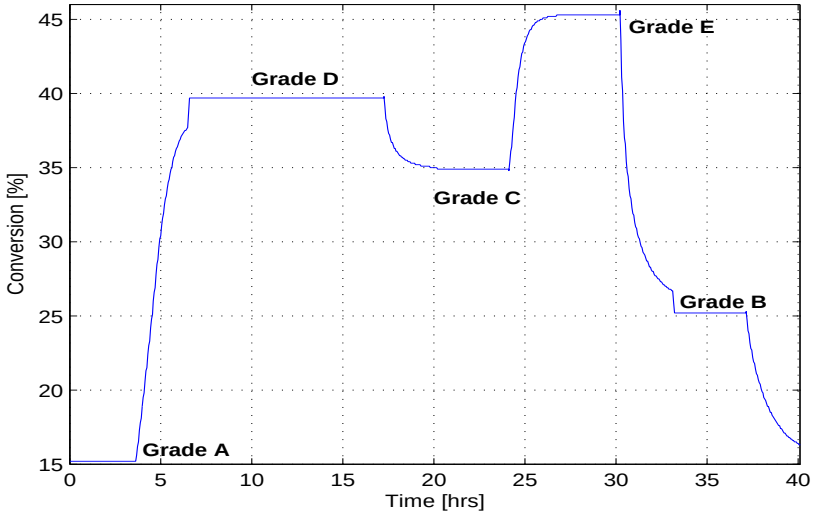


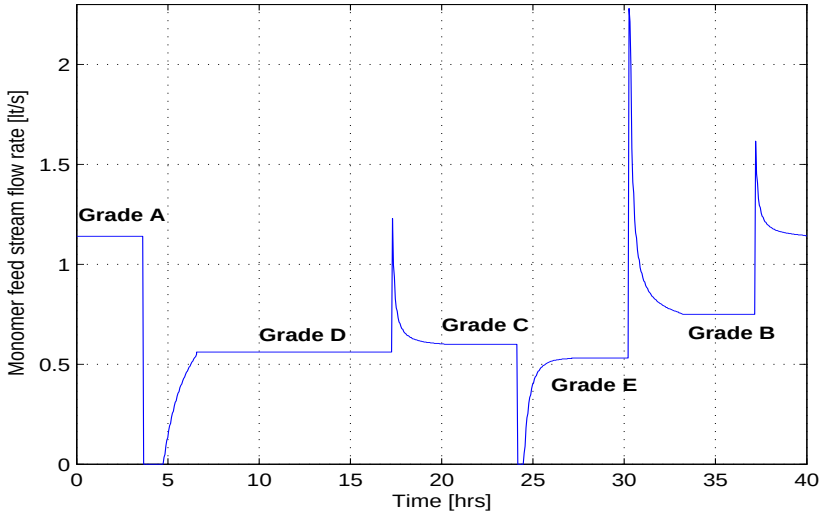


Example: High Impact Polystyrene (HIPS)









Decomposition Optimization Approach to Solve Larger Size MINLP Problems

- Most MINLP solution strategies tend to work well for small to medium size problems
- Some of the best well known MINLP solution strategies solve problems with either : (a) large number of binary variables (but mild nonlinearities) or (b) small number of binary variables (but stronger nonlinearities)
- Hence, normally MINLP codes tend to be unable to solve problems with large number of variables and strong nonlinear behaviour
- Decomposition Optimization techniques can be an efficient way, and sometimes the only way, to solve large scale, highly nonlinear MINLPs

The objective of this section is to solve the Simultaneous Scheduling and Control problem based on our previous formulation by exploiting its decomposable nature through a Lagrangean Decomposition technique. The reformulated model is solved using a decomposition technique and 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. It has been found that this technique can greatly reduce the time spent solving large MINLPs.

Short Tutorial

The main idea behind decomposition methods consists in realizing that in optimization problems the constraints can be divided into "easy" and "hard" constraints.

- Easy constraints are those that are relatively easy to converge (.e.g. linear or quasi-linear constraints)
- Hard constraints are difficult to converge (i.e. non-convex constraints related to nonlinearities)

If the primal optimization problem is decomposed into a series of problems (each one easier to solve than the primal one), then the overall solution of such problem could be easier to achieve. Indeed, due to the embedded nonlinearities and problem size, sometimes decomposition methods can be the only way to solve a given MINLP problem.

Lagrangean Decomposition

Let us assume that we want to solve the following general MINLP:

$$\begin{aligned}
\max \quad & \mathcal{Z}^P = c^T x + d^T (y^1 + y^2) && && \mathbf{P} && (39)\\
\text{s.t.} \quad & A^1 x + B^1 y^1 && \leqslant && b^1 && (40)\\
& A^2 x + B^2 y^2 && \leqslant && b^2 && (41)\\
& h_1(x) && \leqslant && 0 && (42)\\
& h_2(x) && \leqslant && 0 && (43)\\
& x \geqslant 0;\ y^1, y^2 \in \{0,1\} && && && (44)
\end{aligned}$$

Reformulating the problem $\mathbf{P}$

The first thing to do is to "duplicate" the continuous variables (x). We introduce a new variables vector (z) and divide the constraints set into two sets:

- One containing only the variables x
- The other one comprising the variables z

Of course, there will be integer variables in both the x and z constraints set. However, the partition of the constrains set should be done in such a way that the integer variables appearing in the x constraints set should not be contained in the z constraints set. Hence, the reformulated problem reads as,

$$\begin{aligned}
\max \quad & \mathcal{Z}^{RP} = c^T x + d^T (y^1 + y^2) && && \textbf{RP} && (45)\\
\text{s.t.} \quad & A^1 x + B^1 y^1 && \leqslant && b^1 && (46)\\
& A^2 z + B^2 y^2 && \leqslant && b^2 && (47)\\
& h_1(x) && \leqslant && 0 && (48)\\
& h_2(z) && \leqslant && 0 && (49)\\
& x && = && z && (50)\\
& x, z \geqslant 0;\ y^1, y^2 \in \{0,1\} && && && (51)
\end{aligned}$$

The following points about the **RP** formulation should be remarked.

- The integer variables set y^1 appears only in the x constraints set (Eqn 46).
- Similarly, the integer variables set y^2 appears only in the z constraints set (Eqn 47).
- By introducing the constraint $x = z$ (Eqn 50), the **P** and **RP** formulations are totally equivalent.

- Accordingly, the z "duplicated" decision variables also hold the constraint $z \geqslant 0$ (Eqn 51).

Splitting the RP formulation

Now if the constraint $x = z$ is dualized, the **RP** formulation can be written as

$$\begin{aligned}
\max \quad & \mathcal{Z}^{DRP} = c^T x + d^T (y^1 + y^2) + \lambda^T (z - x) && \textbf{DRP} && (52)\\
\text{s.t.} \quad & A^1 x + B^1 y^1 \leqslant b^1 && && (53)\\
& A^2 z + B^2 y^2 \leqslant b^2 && && (54)\\
& h_1(x) \leqslant 0 && && (55)\\
& h_2(z) \leqslant 0 && && (56)\\
& x, z \geqslant 0;\ y^1, y^2 \in \{0,1\} && && (57)
\end{aligned}$$

as it can be easily noted, the above **DRP** formulation can be written (decomposed) into the following two independent formulations.

$$\begin{aligned}
\max \quad & \mathcal{Z}^{DRP_1} = c^T x + d^T y^1 - \lambda^T x && \textbf{DRP}_\textbf{1} && (58)\\
\text{s.t.} \quad & A^1 x + B^1 y^1 \leqslant b^1 && && (59)\\
& h_1(x) \leqslant 0 && && (60)\\
& x \geqslant 0,\ y^1 \in \{0,1\} && && (61)
\end{aligned}$$

$$\max \quad \mathcal{Z}^{DRP_2} = d^T y^2 + \lambda^T z \qquad \textbf{DRP}_\textbf{2} \qquad (62)$$

$$\begin{aligned} \text{s.t.} \quad A^2 z + B^2 y^2 &\leqslant b^2 && (63)\\ h_2(z) &\leqslant 0 && (64)\\ z \geqslant 0,\ y^2 &\in \{0,1\} && (65) \end{aligned}$$

- The decision variables related to the $\mathbf{DRP_1}$ formulation are x and y^1.
- The decision variables related to the $\mathbf{DRP_2}$ formulation are z and y^2.
- Accordingly, the $\mathbf{DRP_1}$ and $\mathbf{DRP_2}$ formulations can be solved independently as MINLPs.

Computing Upper Bounds

When the constraints are convex, the sum of the objective function values of the $\mathbf{DRP_1}$ and $\mathbf{DRP_2}$ formulations are an upper bound on the optimal value of the primal problem $\mathbf{P}$. Thereby, if we denote

$$\hat{\mathcal{Z}} = \mathcal{Z}^{DRP_1} + \mathcal{Z}^{DRP_2} \tag{66}$$

the above statement means that

$$\mathcal{Z}^P \leqslant \hat{\mathcal{Z}} \tag{67}$$

however, if the problem to be solved features non-convexities, then the above inequality will not be necessarily true. In strict terms, computing the smaller upper bound on $\mathcal{Z}^P$ amounts to solve the following Lagrangean dual problem:

$$\mathcal{Z}^D = \min_{\lambda} \ \mathcal{Z}^{DRP} \qquad \mathbf{D} \tag{68}$$

nevertheless, the above $\mathbf{D}$ formulation tends to be difficult to solve. This is the reason why, even when the MINLP problem to be solved might be a non-convex one, the Lagrangean decomposition

approach stills is used for solving MINLPs. Of course, in this case the inequality given by Eqn 67 is only used as an heuristic. Moreover, because of non-convexities, no optimality proof can be offered. Following the computation of a valid lower bound on the $\mathcal{Z}^P$ optimal value is discussed.

Computing Lower Bounds

The lower bound $\underline{\mathcal{Z}}$ is computed by fixing in problem **P** the binary variables and then solving the resulting NLP problem.

Updating Lagrange Multipliers

To generate upper bounds on problem **P**, the Lagrange multipliers λ are computed from the Fisher formula:

$$\lambda^{k+1} = \lambda^k + t^k(y^k - x^k) \tag{69}$$

$$t^{k+1} = \frac{\alpha_k(LD(\lambda^k) - P^*)}{||y^k - x^k||^2} \tag{70}$$

where k stands for iteration number, t^k is a scalar step size and α_k is a scalar variable which is normally constrained between [0,2], but it can be decreased to improve convergence.

Example

The application of the Lagrangean decomposition approach for solving MINLPs is shown using the following example:

$$\begin{aligned}
\max \quad & \mathcal{Z}^P = -(5y_1 + 6y_2 + 8y_3 + 10x_1 - 7x_6 - 18\ln(1+x_2) \\
& \qquad -19.2\ln(1+x_1-x_2) + 10) && (71)\\
\text{s.t.} \quad & 0.8\ln(1+x_2) + 0.96\ln(1+x_1-x_2) - 0.8x_6 \geqslant 0 && (72)\\
& x_2 - x_1 \leqslant 0 && (73)\\
& x_2 - 2y_1 \leqslant 0 && (74)\\
& x_1 - x_2 - 2y_2 \leqslant 0 && (75)\\
& \ln(1+x_2) + 1.2\ln(1+x_1-x_2) - x_6 - 2y_3 \geqslant -2 && (76)\\
& y_1 + y_2 \leqslant 1 && (77)\\
& x_1, x_2, x_6 \geqslant 0;\ y_1, y_2, y_3 \in \{0,1\}
\end{aligned}$$

The solution of this problem is reported as:

$$\begin{aligned}
\mathcal{Z}^* &= 5.5796\\
x_1^* &= 1.76\\
x_2^* &= 0\\
x_3^* &= 1.218\\
y_1^* &= 0\\
y_2^* &= 1\\
y_3^* &= 0
\end{aligned}$$

The first step aims to write the primal problem as a reformulated one by the introduction of cloned variables z_1, z_2 and z_6. Following we have to divide the primal problem into two constraint sets:

- One of them should contain the x variables vector and some binary variables
- The other one should comprise the z variables vector and the remaining binary variables

Recall that the two sets of constraints should be comprised of different binary variables (e.g. binary variables associated to the x variables constraint set cannot be a member of the z variables vector and viceversa). If we have a close look at the above formulation, we can notice that constraint 77 dictates that the y_1 and y_2 binary variables should appear together since they are related trough the inequality

$$y_1 + y_2 \leqslant 1$$

therefore all the constraints featuring either y_1 and/or y_2 should be part of one of the constraint sets into which the primal problem will be divided. Therefore, the first set of constraints will feature the y_1 and y_2 binary variables and is given as follows.

$$\begin{aligned} x_2 &\leqslant 2y_1 \\ x_1 - x_2 &\leqslant 2y_2 \\ y_1 + y_2 &\leqslant 1 \\ 0.8\ln(1+x_2) + 0.96\ln(1+x_1-x_2) - 0.8x_6 &\geqslant 0 \end{aligned}$$

The second set of constraints will feature the remaining y_3 binary variables and additional constraints not included in the above set. Hence,

$$\begin{aligned} z_2 - z_1 &\leqslant 0 \\ \ln(1+z_2) + 1.2\ln(1+z_1-z_2) - z_6 - 2y_3 &\geqslant -2 \end{aligned}$$

You should notice that in partitioning the constraints set, we have decided to include the constraint involving logarithmic (and no binary variables) terms in the first set of constraints and the other constraint involving logarithmic terms and the y_3 binary variable into the second constraints set.

Thereby, the reformulated problem reads,

$$
\begin{aligned}
\max \quad & \mathcal{Z}^{RP} = -(5y_1 + 6y_2 + 8y_3 + 10x_1 - 7x_6 - 18\ln(1 + x_2) \\
& \qquad -19.2\ln(1 + x_1 - x_2) + 10) \\
\text{s.t.} \quad & x_2 \leqslant 2y_1 \\
& x_1 - x_2 \leqslant 2y_2 \\
& y_1 + y_2 \leqslant 1 \\
& 0.8\ln(1 + x_2) + 0.96\ln(1 + x_1 - x_2) - 0.8x_6 \geqslant 0 \\
& z_2 - z_1 \leqslant 0 \\
& \ln(1 + z_2) + 1.2\ln(1 + z_1 - z_2) - z_6 - 2y_3 \geqslant -2 \\
& x_1 = z_1 \\
& x_2 = z_2 \\
& x_6 = z_6 \\
& x_1, x_2, x_6, z_1, z_2, z_6 \geqslant 0;\ y_1, y_2, y_3 \in \{0, 1\}
\end{aligned}
$$

Now if the above formulation is dualized:

$$
\begin{aligned}
\max \quad & \mathcal{Z}^{DRP} = -(5y_1 + 6y_2 + 8y_3 + 10x_1 - 7x_6 - 18\ln(1 + x_2) \\
& -19.2\ln(1 + x_1 - x_2) + 10 + \lambda^1(z_1 - x_1) + \lambda^2(z_2 - x_2) + \lambda^6(z_6 - x_6)) \\
\text{s.t.} \quad & x_2 \leqslant 2y_1 \\
& x_1 - x_2 \leqslant 2y_2 \\
& y_1 + y_2 \leqslant 1 \\
& 0.8\ln(1 + x_2) + 0.96\ln(1 + x_1 - x_2) - 0.8x_6 \geqslant 0 \\
& z_2 - z_1 \leqslant 0 \\
& \ln(1 + z_2) + 1.2\ln(1 + z_1 - z_2) - z_6 - 2y_3 \geqslant -2 \\
& x_1, x_2, x_6, z_1, z_2, z_6 \geqslant 0; \;\; y_1, y_2, y_3 \in \{0, 1\}
\end{aligned}
$$

Finally the above formulation can be cast in terms of the following two independent formulations:

$$
\begin{aligned}
\max \quad & \mathcal{Z}^{DRP_1} = -(5y_1 + 6y_2 + 8y_3 + 10x_1 - 7x_6 - 18\ln(1 + x_2) \\
& -19.2\ln(1 + x_1 - x_2) + 10 - \lambda^1 x_1 - \lambda^2 x_2 - \lambda^6 x_6) \\
\text{s.t.} \quad & x_2 \leqslant 2y_1 \\
& x_1 - x_2 \leqslant 2y_2 \\
& y_1 + y_2 \leqslant 1 \\
& 0.8\ln(1 + x_2) + 0.96\ln(1 + x_1 - x_2) - 0.8x_6 \geqslant 0 \\
& x_1, x_2, x_6 \geqslant 0; \;\; y_1, y_2 \in \{0, 1\}
\end{aligned}
$$

and,

$$
\begin{array}{lrcl}
\max & \mathcal{Z}^{DRP_2} = -(8y_3 + \lambda^1 z_1 + \lambda^2 z_2 + \lambda^6 z_6) & & \\
\text{s.t.} & z_2 - z_1 & \leqslant & 0 \\
& \ln(1+z_2) + 1.2\ln(1+z_1-z_2) - z_6 - 2y_3 & \geqslant & -2 \\
& z_1, z_2, z_6 \geqslant 0;\ y_3 \in \{0,1\} & &
\end{array}
$$

Gams Code

```
$title Simple MINLP Problem (Problem No.1 from Marco Duran PhD Thesis)
* ---------------------------------------------------------------
* A Lagrangean Decomposition Approach for Solving MINLPs
*
* Written by Antonio Flores T.
* 9 March, 2006
* ---------------------------------------------------------------
*
Variables  profit,x1,x2,x6,z1,z2,z6,lambda1_dummy,lambda2_dummy,lambda6_dummy ;
Variables  profit_z1, profit_z2,zlow;

Binary variables  y1,y2,y3 ;

Equations         obj,r1,r2,r3,r4,r5,r6 ;
Equations         objz1,objz2;
Equations         objlower,r7,r8,r9,r10,r11,r12;

Scalar            alpha /1/;

parameter         zupper,zlower,niters,maxniters;
parameters        lambda1,lambda2,lambda6;
parameters        diff1,diff2,diff6,errnorm;
parameters        y1fixed, y2fixed,y3fixed;
parameters        tk,lambda1_old,lambda2_old,lambda6_old;

zupper            = inf;
zlower            = -inf;
```

```
niters             = 0;
maxniters          = 5;

* ---------------------------------------------------------------------------
* Form the Reformulated Problem (RP) from which a relaxed MINLP solution is computed
* ---------------------------------------------------------------------------

obj .. profit =e= -(5*y1+6*y2+8*y3+10*x1-7*x6-18*log(1+x2)-19.2*log(1+x1-x2)+10
                   +lambda1_dummy*(z1-x1)+lambda2_dummy*(z2-x2)+lambda6_dummy*(z6-x6)) ;

r1.. 0.8*log(1+x2)+0.96*log(1+x1-x2)-0.8*x6 =g= 0 ;

r2.. z2-z1 =l= 0 ;

r3.. x2-2*y1 =l= 0 ;

r4.. x1-x2-2*y2 =l= 0 ;

r5.. log(1+z2)+1.2*log(1+z1-z2)-z6-2*y3 =g= -2 ;

r6.. y1+y2 =l= 1;

x1.lo = 0 ;
x2.lo = 0 ;
x6.lo = 0 ;
z1.lo = 0 ;
z2.lo = 0 ;
z6.lo = 0 ;
```

```
lambda1_dummy.lo = 0;
lambda1_dummy.up = 5;
lambda2_dummy.lo = 0;
lambda2_dummy.up = 5;
lambda6_dummy.lo = 0;
lambda6_dummy.up = 5;

model RP /obj,r1,r2,r3,r4,r5,r6 / ;

* ---------------------------------------------------------------------
* Form the two indepedent MINLPs into which the RP has been decomposed
* ---------------------------------------------------------------------

objz1.. profit_z1 =e= -(5*y1+6*y2+10*x1-7*x6-18*log(1+x2)-19.2*log(1+x1-x2)+10
                      -lambda1*x1-lambda2*x2-lambda6*x6) ;

model LRP_1 /objz1,r1,r3,r4,r6/ ;

objz2.. profit_z2 =e= -(8*y3+lambda1*z1+lambda2*z2+lambda6*z6);

model LRP_2 /objz2,r2,r5/ ;

* ---------------------------------------------------------------------
* By fixing the binary variables into the RP problem, compute a lower bound
* on the optimal value of the original objective function solving a NLP
* ---------------------------------------------------------------------

objlower .. zlow   =e= -(5*y1fixed+6*y2fixed+8*y3fixed+10*x1-7*x6-18*log(1+x2)
```

```
                    -19.2*log(1+x1-x2)+10);

r7.. x2 - 2*y1fixed =l= 0 ;

r8.. x1-x2-2*y2fixed =l= 0 ;

r9.. log(1+z2)+1.2*log(1+z1-z2)-z6-2*y3fixed =g= -2 ;

r10.. x1-z1 =e= 0 ;

r11.. x2-z2 =e= 0 ;

r12.. x6-z6 =e= 0 ;

model LRP_LB /objlower,r1,r2,r7,r8,r9,r10,r11,r12/ ;

* -------------------------------------------------------------
* Beginning of the iterative Lagrangean Decomposition procedure
* -------------------------------------------------------------

solve RP maximizing profit using rminlp ;
lambda1 = lambda1_dummy.L;
lambda2 = lambda2_dummy.L ;
lambda6 = lambda6_dummy.L ;

while ( (zlower lt zupper),
      niters = niters+1;
*
```

```
* Compute an optimal value upper bound
*
      solve LRP_1 maximizing profit_z1 using minlp ;
      solve LRP_2 maximizing profit_z2 using minlp ;
      errnorm   = sqr(z1.L-x1.L)+sqr(z2.L-x2.L)+sqr(z6.L-x6.L) ;
      diff1     = z1.L-x1.L ;
      diff2     = z2.L-x2.L ;
      diff6     = z6.L-x6.L ;
*
* Fixing binary variable to get an optimal value lower bound
*
      y1fixed = y1.L ;
      y2fixed = y2.L ;
      y3fixed = y3.L ;
      solve LRP_LB maximizing zlow using nlp ;
* ------------------------------------------------------
* Update Lagrange multipliers by a simple rule
* ------------------------------------------------------
      lambda1_old = lambda1;
      lambda2_old = lambda2;
      lambda6_old = lambda6;
      zupper      = profit_z1.L+profit_z2.L ;
      zlower      = zlow.L ;
      tk          = alpha*(zupper-zlower)/errnorm ;
      lambda1     = lambda1_old+tk*diff1;
      lambda2     = lambda2_old+tk*diff2;
      lambda6     = lambda6_old+tk*diff6;
```

```
      display lambda1,lambda2,lambda6,tk,zlower,zupper;
);
*-- End of the ejemplo-1-relaxation.gms file --
```

- Objective Function

$$\max \quad \left\{ \sum_{i=1}^{N_p} \frac{C_i^p W_i}{T_c} - \sum_{i=1}^{N_p} \frac{C_i^s (G_i - W_i/T_c)\Theta_i}{2} \right.$$
$$- \left[\sum_{k=1}^{N_s} \sum_{f=1}^{N_f^e} h_{fk} \theta_k^t Q_{max}^m \sum_{c=1}^{N_{cp}} u_{fck}^m \gamma_c \right] \frac{C^r}{T_c}$$
$$\left. - \left[\sum_{k=1}^{N_s} \sum_{f=1}^{N_f^e} h_{fk} \theta_k^t Q_{max}^I \sum_{c=1}^{N_{cp}} u_{fck}^I \gamma_c \right] \frac{C^I}{T_c} \right\} \tag{78}$$

- Initial and final controlled and manipulated variable values at each slot

$$x_{in,k}^n = \sum_{i=1}^{N_p} x_{ss,i}^n y_{i,k}, \quad \forall n, k \tag{79}$$

$$\bar{x}_k^n = \sum_{i=1}^{N_p} x_{ss,i}^n y_{i,k+1}, \quad \forall n, k \neq N_s \tag{80}$$

$$\bar{x}_k^n = \sum_{i=1}^{N_p} x_{ss,i}^n y_{i,1}, \quad \forall n, k = N_s \tag{81}$$

$$u_{in,k}^m = \sum_{i=1}^{N_p} u_{ss,i}^m y_{i,k}, \quad \forall m, k \tag{82}$$

$$\bar{u}_k^m = \sum_{i=1}^{N_p} u_{ss,i}^m y_{i,k+1}, \quad \forall m, k \neq N_s - 1 \tag{83}$$

$$\bar{u}_k^m = \sum_{i=1}^{N_p} u_{ss,i}^m y_{i,1}, \quad \forall m, k = N_s \tag{84}$$

$$u_{1,1,k}^m = u_{in,k}^m, \quad \forall m, k \tag{85}$$

$$x_{o,1,k}^n = x_{in,k}^n, \quad \forall n, k \tag{86}$$

$$x_{tol,k}^n \geqslant x_{Nfe,Nc,k}^n - \bar{x}_k^n, \quad \forall n, k \tag{87}$$

$$-x_{tol,k}^n \leqslant x_{Nfe,Nc,k}^n - \bar{x}_k^n, \quad \forall n, k \tag{88}$$

- Smooth transition constraints

$$u^{m}_{f,c,k} - u^{m}_{f,c-1,k} \leqslant u^{c}_{cont}, \ \forall m,k,c \neq 1 \quad (89)$$

$$u^{m}_{f,c,k} - u^{m}_{f,c-1,k} \geqslant -u^{c}_{cont}, \ \forall m,k,f,c \neq 1 \quad (90)$$

$$u^{m}_{f,1,k} - u^{m}_{f-1,Nfe,k} \leqslant u^{f}_{cont}, \ \forall m,k,f \neq 1 \quad (91)$$

$$u^{m}_{f,1,k} - u^{m}_{f-1,Nfe,k} \geqslant -u^{f}_{cont}, \ \forall m,k,f \neq 1 \quad (92)$$

$$u^{m}_{1,1,k} - u^{m}_{in,k} \leqslant u^{f}_{cont}, \ \forall k \quad (93)$$

$$u^{m}_{1,1,k} - u^{m}_{in,k} \geqslant -u^{f}_{cont}, \ \forall k \quad (94)$$

$$\dot{x}^{n}_{Nfe,Ncp,k} \geqslant -\dot{x}_{tol,k}, \ \forall n,k \quad (95)$$

$$\dot{x}^{n}_{Nfe,Ncp,k} \leqslant \dot{x}_{tol,k}, \ \forall n,k \quad (96)$$

Lagrangean Decomposition

In a Lagrangean Decomposition technique certain variables are duplicated and set equal by new constraints. These new constraints are then relaxed through Lagrangean Relaxation, yielding a decomposable model over two or more subsets of constraints

Consider the following mathematical programming problem:

$$(\mathrm{P}) \quad \max\left\{\mathrm{fx}|\mathrm{Ax} \leqslant \mathrm{b}, \mathrm{Cx} \leqslant \mathrm{d}, \mathrm{x} \in \mathrm{X}\right\}$$

which is equivalent to:

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

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

$$(\mathrm{LDu})$$

$$\max\left\{\mathrm{fx} + \mathrm{u}(\mathrm{y} - \mathrm{x})|\mathrm{Cx} \leqslant \mathrm{d}, \mathrm{x} \in \mathrm{X}, \mathrm{Ay} \leqslant \mathrm{b}, \mathrm{y} \in \mathrm{Y}\right\} \tag{97}$$

$$= \max\left\{(\mathrm{f} - \mathrm{u})\,\mathrm{x}|\mathrm{Cx} \leqslant \mathrm{d}, \mathrm{x} \in \mathrm{X}\right\} + \max\left\{\mathrm{uy}|\mathrm{Ay} \leqslant \mathrm{b}, \mathrm{y} \in \mathrm{Y}\right\} \tag{98}$$

If the feasible regions are convex, then LDu is an upper bound for P for any given u. 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:

$$\mathrm{P} = \min_{\mathrm{u}} \mathrm{LDu}$$

In the presence of integer variables and other nonconvexities a duality gap may exist. Since this is the case of the current formulation, the search for an optimum will be performed using an heuristic approach 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:

$$\mathrm{u}^{\mathrm{k}+1} = \mathrm{u}^{\mathrm{k}} + \mathrm{t}^{\mathrm{k}}(\mathrm{y}^{\mathrm{k}} - \mathrm{x}^{\mathrm{k}}), \quad \mathrm{t}^{\mathrm{k}+1} = \frac{\alpha_{\mathrm{k}}(\mathrm{LD}(\mathrm{u}^{\mathrm{k}}) - \mathrm{P}^{*})}{\| \mathrm{y}^{\mathrm{k}} - \mathrm{x}^{\mathrm{k}} \|^{2}} \tag{99}$$

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.

The problem reformulation consists of four steps.

1. Duplicate key variables.

$$z_{ik} = y_{ik}, \forall i, k, y \in B, z \in CO(B) \quad (100a)$$
$$\phi_k^t = \theta_k^t, \forall k \quad (100b)$$
$$Dc = Tc \quad (100c)$$

where B is the set of binary values {0,1} and CO(B) is the Convex Hull of set B.

Equations 100a to 100c 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 ($z \in CO(B)$).

2. Assign one copy of each variable to the scheduling constraints and the other to the dynamic optimization constraints; add necessary extra constraints.
   - ϕ_k^t substitutes θ_k^t in equations 24, 24 and 26.
   - The following two equations are duplicates of 2a and 2b:

$$\sum_{k=1}^{N_s} z_{ik} = 1, \ \forall i \tag{101a}$$

$$\sum_{i=1}^{N_p} z_{ik} = 1, \ \forall k \tag{101b}$$

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

3. Equations 100a,100b, and 100c are eliminated and added to the objective function by means of a Lagrangian Relaxation.

   The objective function takes the following form:

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

$$- \left[\sum_{k=1}^{N_s} \sum_{f=1}^{N_f e} h_{fk} \theta_k^t Q_{max}^I \sum_{c=1}^{N_{cp}} u_{fck}^I \gamma_c \right] \frac{C^I}{D_c}$$

$$+ \sum_{k=1}^{N_s} \sum_{i=1}^{N_p} \left[\mu_y (z_{ik} - y_{ik}) \right] + \sum_{k=1}^{N_s} \left[\mu_\theta (\phi_k^t - \theta_k^t) \right] + \mu_{Tc} (D_c - T_c) \Bigg\}$$

$\mu_y, \mu_\theta, \mu_{Tc}$ 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.

$$\max \quad \left\{ \sum_{i=1}^{N_p} \frac{C_i^p W_i}{T_c} - \sum_{i=1}^{N_p} \frac{C_i^s (G_i - W_i/T_c)\theta_i}{2} \right.$$

$$\left. + \sum_{k=1}^{N_s} \sum_{i=1}^{N_p} \left[\mu_y (-y_{ik}) \right] + \sum_{k=1}^{N_s} \left[\mu_\theta (-\theta_k^t) \right] + \mu_{Tc} (-T_c) \right\} \qquad (102)$$

$$s.t.$$

Equations. 2a to 6e

- Control Subproblem

$$\max \quad \left\{ -\left[\sum_{k=1}^{N_s}\sum_{f=1}^{N_f e} h_{fck}\theta_k^t Q_{max}^I \sum_{c=1}^{N_{cp}} u_{fck}^I \gamma_c\right]\frac{C^I}{D_c} \right.$$
$$-\left[\sum_{k=1}^{N_s}\sum_{f=1}^{N_f e} h_{fk}\theta_k^t Q_{max}^m \sum_{c=1}^{N_{cp}} u_{fck}^m \gamma_c\right]\frac{C^m}{D_c}$$
$$\left. +\sum_{k=1}^{N_s}\sum_{i=1}^{N_p}\left[\mu_y(z_{ik})\right] + \sum_{k=1}^{N_s}\left[\mu_\theta(\phi_k^t)\right] + \mu_{Tc}(Dc) \right\} \tag{103}$$

$s.t.$

Equations. 6f, 79 to 96, 101a and 101b
with variable y_{ik} substituted for z_{ik}
and variable θ_k^t substituted for ϕ_k^t

Outline of the Solution Strategy

1. Solve relaxed scheduling subproblem (equation 102) and optimal control subproblem (equation 103) with multipliers set to zero.

2. Initialize the Lagrange multipliers using subgradient method (see step 7).

3. Solve scheduling subproblem (equation 102) and optimal control subproblem (equation 103).

4. Obtain an upper bound as scheduling subproblem (equation 102) solution, plus optimal control subproblem (equation 103) solution.

5. Fix the values of the binary variables using the solution to the scheduling subproblem (equation 102). Solve the original problem (with fixed binaries) to obtain lower bound.

6. If $|$ upper bound $-$ lower bound $| \leqslant$ 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:

$$u^{k+1} = u^k + t^k(y^k - x^k), \quad t^{k+1} = \frac{\alpha_k(LD(u^k) - P^*)}{\| y^k - x^k \|^2} \tag{104}$$

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

Case Study: Single HIPS CSTR

Steady States and grade information of the HIPS polymerization reaction train

	Grade N	Grade A	Grade B	Grade 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_m [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

Direct and Lagrangan solutions for HIPS example

Algorithm	Obj. [$/hr]	Opt. Sequence	Cycle [h]	Trans. [h]	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

Problem Size for Direct and Decomposed Solution

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

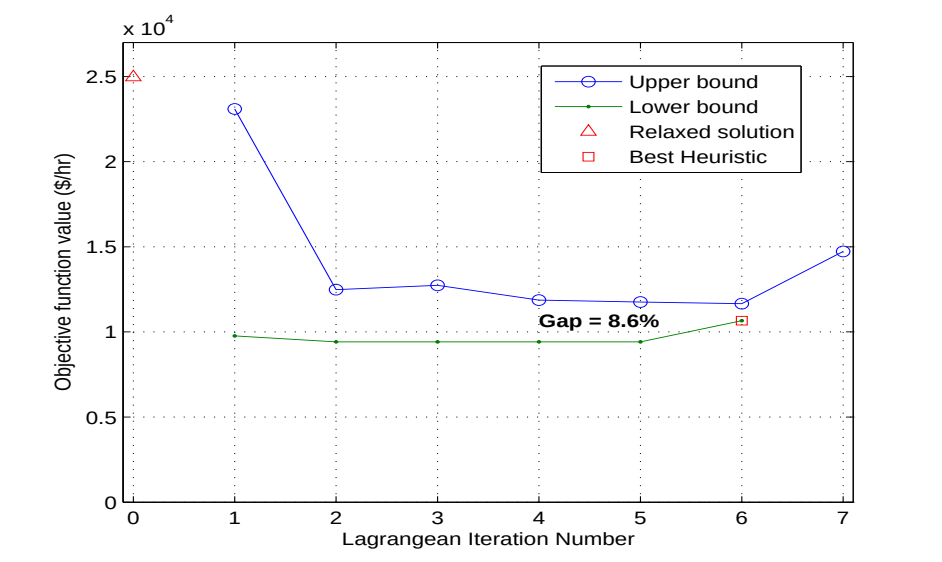
Full space solution. Profit= $ 10500, Cycle time= 88.4 h

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

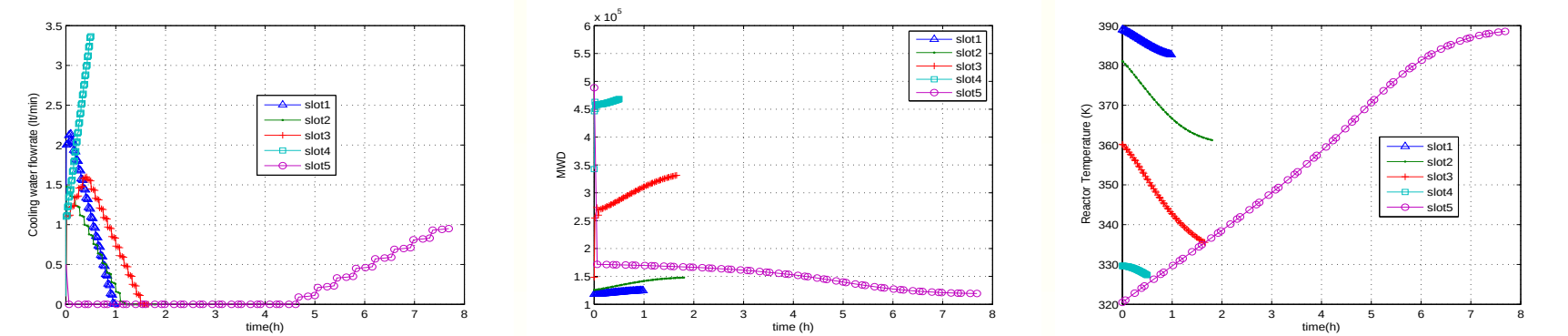
Decomposition heuristic solution. Profit= $ 10568, Cycle time= 87.1 h

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

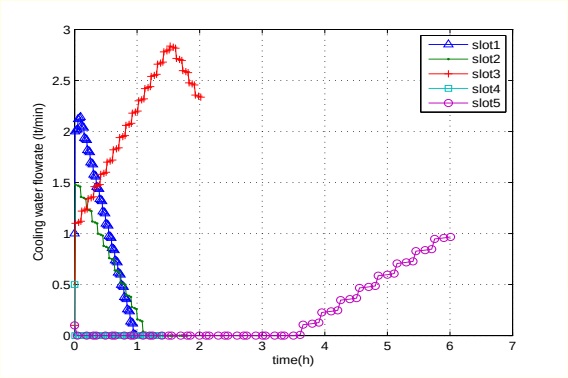
Upper and Lower bounds during Lagrangean Heuristics

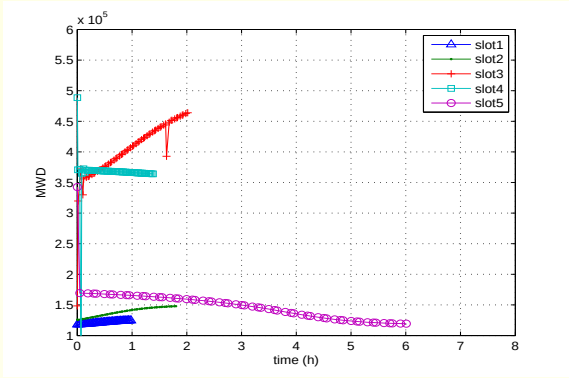


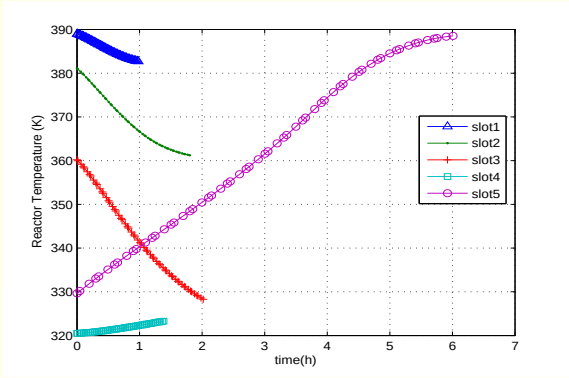
Dynamic transitions obtained by direct solution



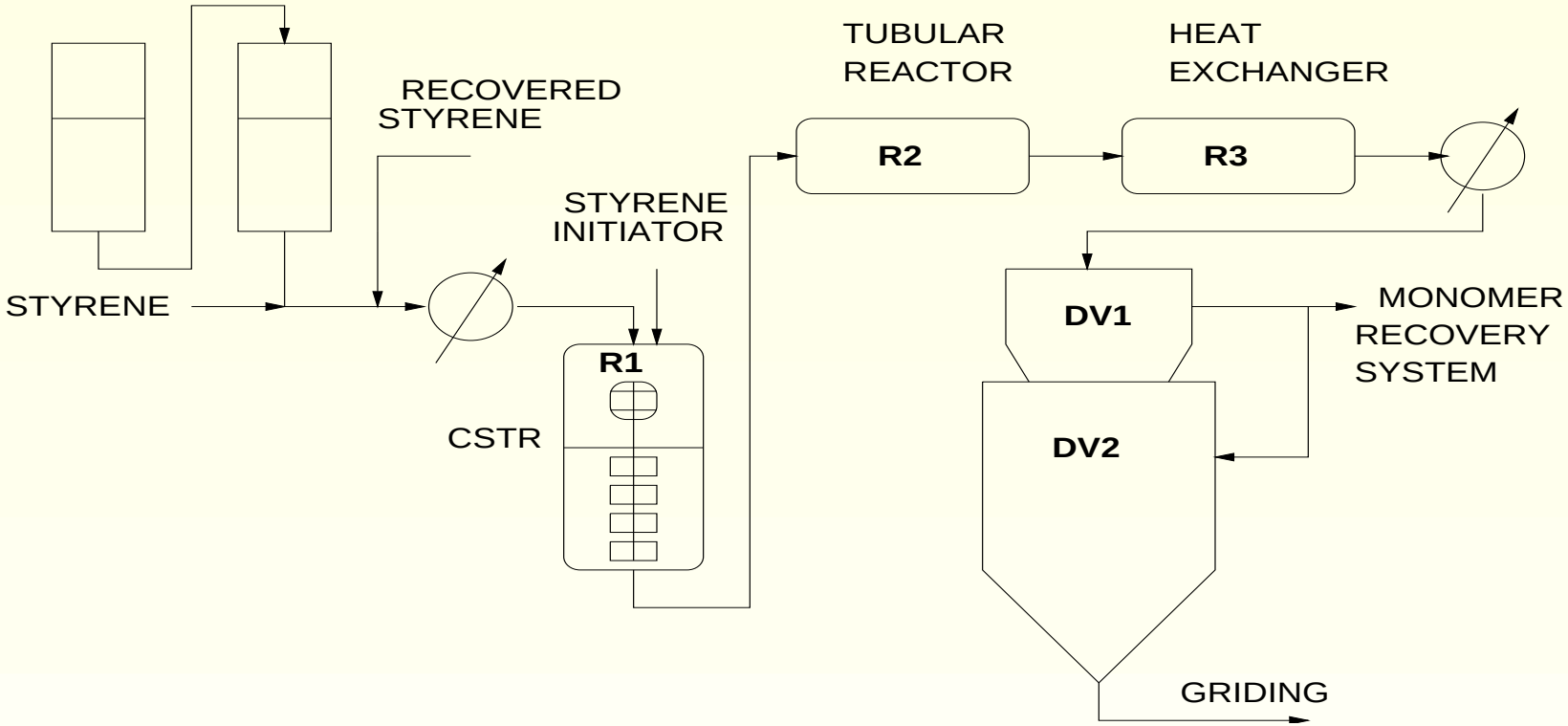
Dynamic transitions obtained by Lagrange Heuristic


Cooling water flowrate (l/min)
time(h)
slot1
slot2
slot3
slot4
slot5

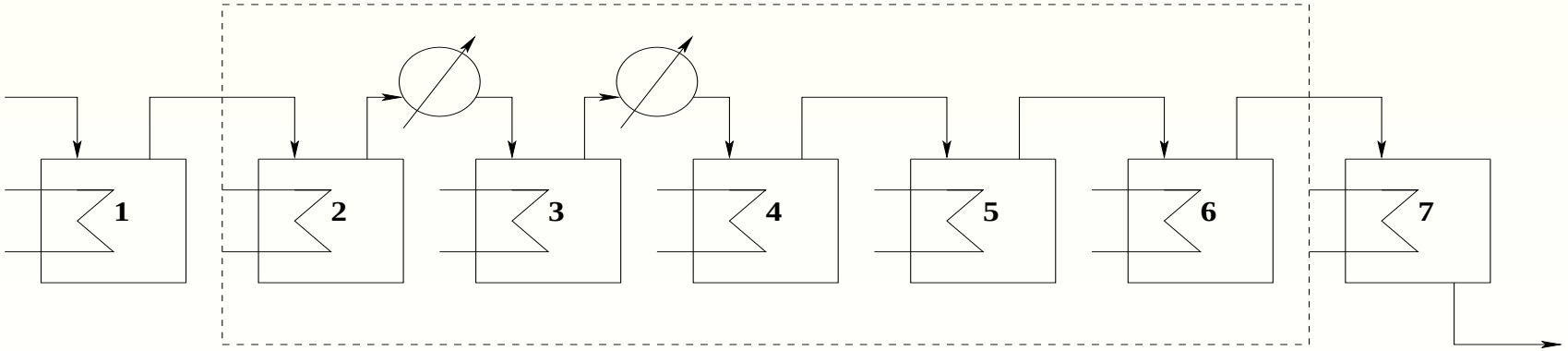

x 10^5
MWD
time (h)
slot1
slot2
slot3
slot4
slot5


Reactor Temperature (K)
time(h)
slot1
slot2
slot3
slot4
slot5

Case Study: HIPS Reaction Train



Approximation of HIPS plant by a set of 7 CSTRs



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

Steady States and grade information of the HIPS reaction train

	Grade N	Grade A	Grade B	Grade 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_m [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

Lagrangean solution for HIPS train example

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

decomposition heuristic results. Profit= $6245, Cycle time=39.2 h

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

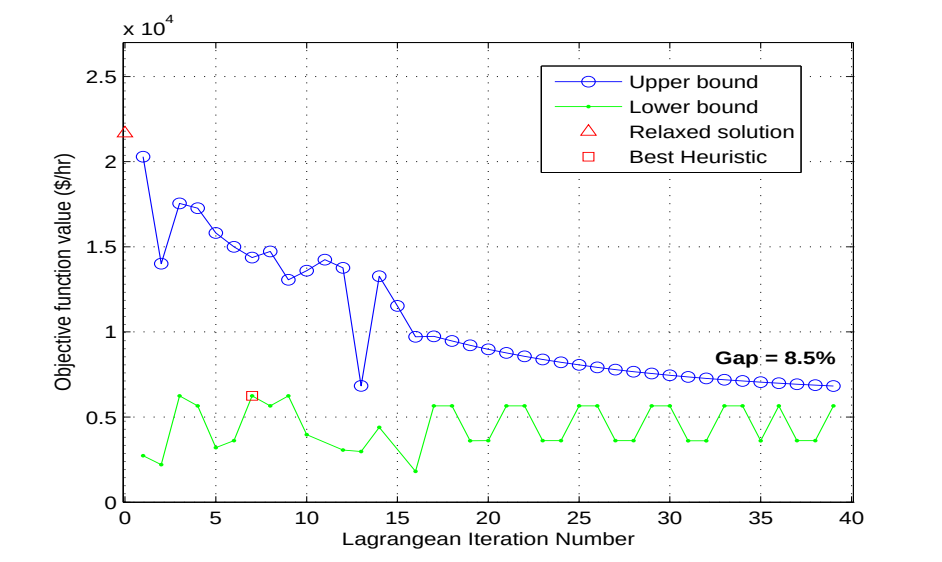
Problem Size for Direct and Decomposed Solution

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

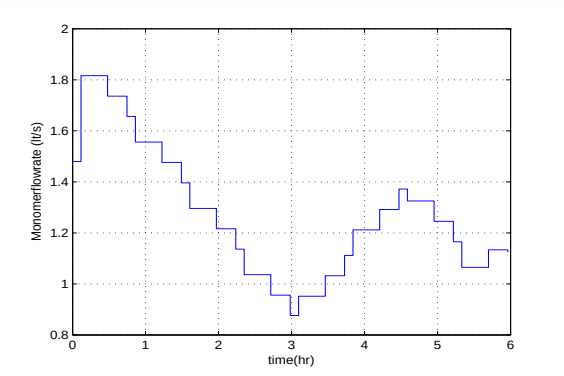
Value of the duplicated variables in the last Lagrangean iteration

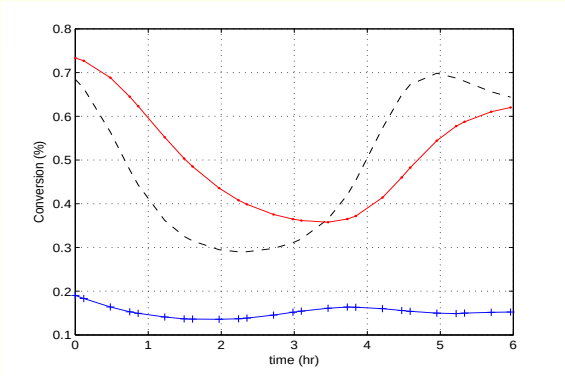
Variable	Scheduling Subproblem	Control Subproblem
y_{B1}	1	-
y_{A2}	1	-
y_{N3}	1	-
y_{C4}	1	-
z_{B1}	-	0.994
z_{A2}	-	0.994
z_{N3}	-	1.000
z_{C4}	-	1.000
z_{A1}	-	0.006
z_{B2}	-	0.006
θ_1^t	0.50	-
θ_2^t	0.50	-
θ_3^t	0.50	-
θ_4^t	0.50	-
ϕ_1^t	-	1.43
ϕ_2^t	-	6.60
ϕ_3^t	-	3.31
ϕ_4^t	-	1.43
Tc	16.634	-
Dc	443.486	-

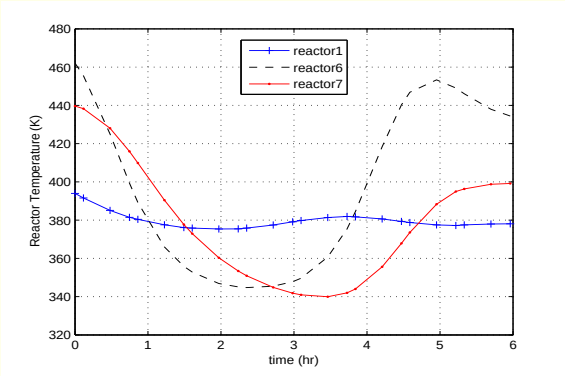
Upper and Lower bounds during Lagrangean Heuristics



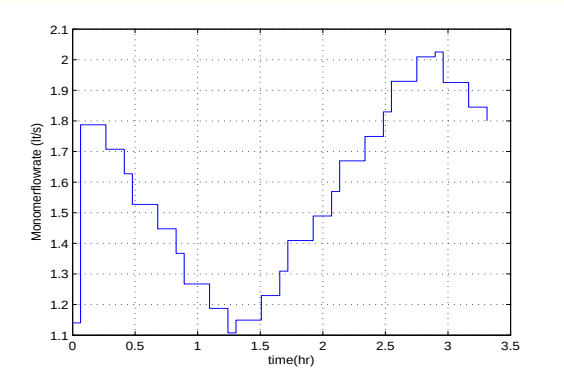
Monomer feed stream, conversion and temperature profiles of reactors 1,6 and 7 in slot 1

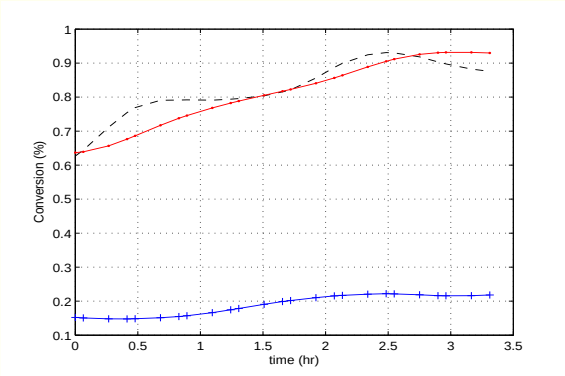


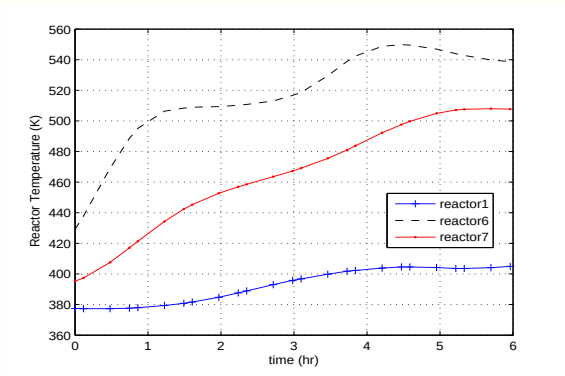




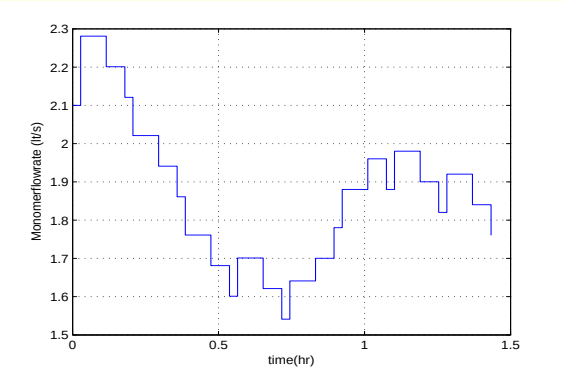
Monomer feed stream, conversion and temperature profiles of reactors 1,6 and 7 in slot 2

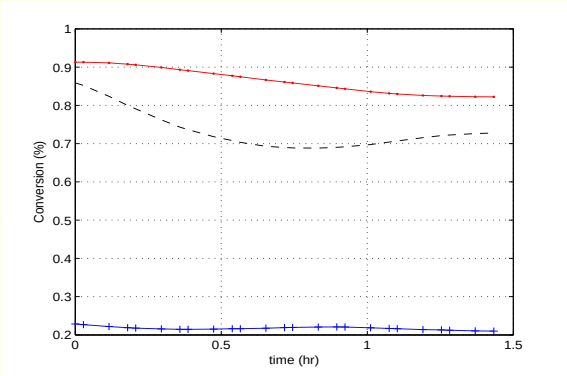


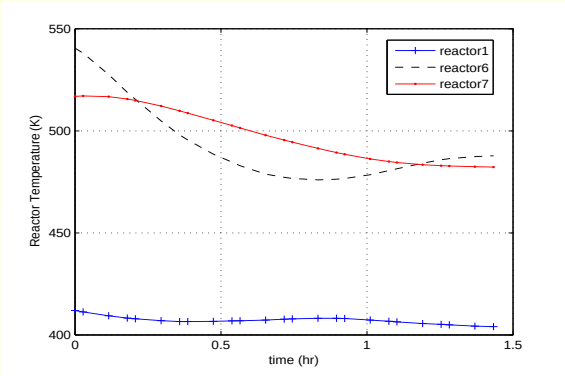




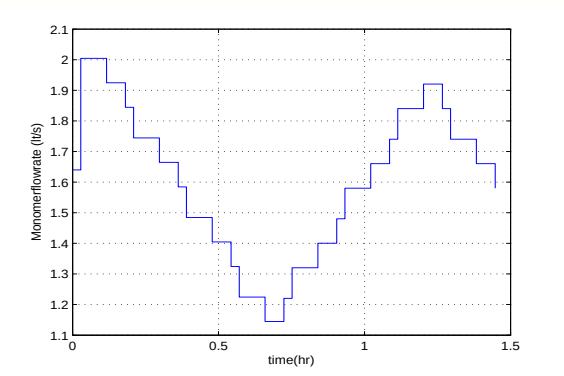
Monomer feed stream, conversion and temperature profiles of reactors 1,6 and 7 in slot 3

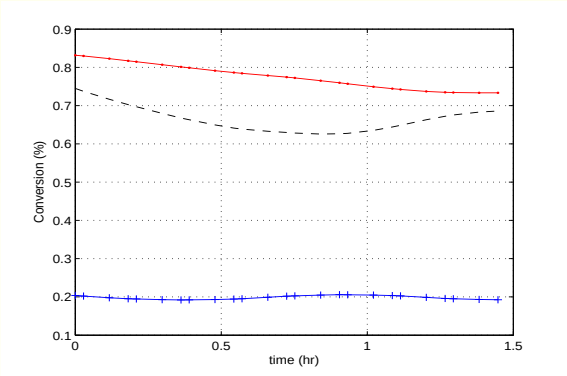


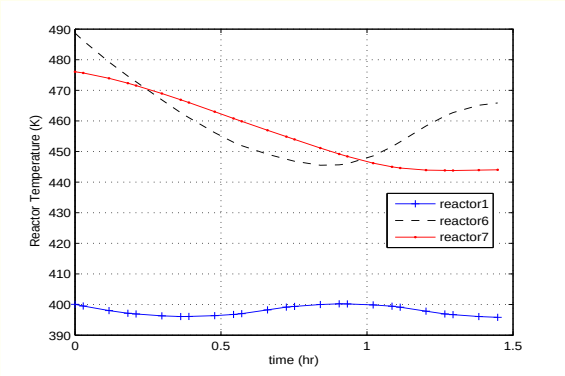




Monomer feed stream, conversion and temperature profiles of reactors 1,6 and 7 in slot 4







Conclusions

- In this work we have addressed the simultaneous cyclic scheduling and control problem for several multiproduct CSTRs. Rather than assuming constant transition times and neglecting process dynamics, a mathematical model, able to describe dynamic process behavior during product transition, was embedded into the optimization formulation. Solving the scheduling and control problem taking into account process dynamics is the rigorous way to address scheduling problems.

- Even in face of nonlinear behavior, the proposed simultaneous cyclic scheduling and control formulation was able to find optimal production sequences. However, convergence towards the optimal solution turned out to be harder to achieve as the nonlinearity of the system increased. Moreover, the presence of nonlinearities creates nonconvexities in the optimization formulation probably leading to obtain suboptimal solutions.

- The Lagrangean Decomposition methodology as presented by Guignard and Kim was used to reformulate the simultaneous scheduling and control problem. 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 scheduling and control problem in the HIPS polymerization system. The computational effort required by the decomposition heuristic is lower than the computational effort required by the direct solution (solution in full space). The Complete HIPS problem was only solvable using the decomposition heuristic.

- The present work does not prove that the solution to the scheduling and control problem in the HIPS reactor train example cannot be obtained without the decomposition. However, it does show is that if such a direct solution is available, the effort required to obtain it becomes unpractical.

Future Work

- Scheduling and Control in Parallel Plants
- Scheduling and Control of Distributed Parameter Systems
- Scheduling and Control of Batch and Semibatch Plants
- Simultaneous Planning, Scheduling and Control