# Carnegie Mellon University

## CARNEGIE INSTITUTE OF TECHNOLOGY

## THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Doctor of Philosophy

TITLE Integration of Scheduling and Dynamic Optimization:
Computational Strategies and Industrial Applications

PRESENTED BY Yisu Nie

ACCEPTED BY THE DEPARTMENT OF

Chemical Engineering

| | |
|---|---|
| LORENZ BIEGLER | 6/23/14 |
| LORENZ BIEGLER, ADVISOR AND DEPARTMENT HEAD | DATE |

APPROVED BY THE COLLEGE COUNCIL

| | |
|---|---|
| VIJAYAKUMAR BHAGAVATULA | 6/23/14 |
| DEAN | DATE |

CARNEGIE MELLON UNIVERSITY

# INTEGRATION OF SCHEDULING AND DYNAMIC OPTIMIZATION: COMPUTATIONAL STRATEGIES AND INDUSTRIAL APPLICATIONS

**A DISSERTATION**

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

*for the degree of*

DOCTOR OF PHILOSOPHY

*in*

CHEMICAL ENGINEERING

*by*

YISU NIE

B.S., CONTROL SCIENCE AND ENGINEERING, ZHEJIANG UNIVERSITY

Pittsburgh, Pennsylvania
June, 2014

*To my family*

# Abstract

This thesis study focuses on the development of model-based optimization strategies for the integration of process scheduling and dynamic optimization, and applications of the integrated approaches to industrial polymerization processes. The integrated decision-making approaches seek to explore the synergy between production schedule design and process unit control to improve process performance. The integration problem has received much attention from both the academia and industry since the past decade. For scheduling, we adopt two formulation approaches based on the state equipment network and resource task network, respectively. For dynamic optimization, we rely on the simultaneous collocation strategy to discretize the differential-algebraic equations. Two integrated formulations are proposed that result in mixed discrete/dynamic models, and solution methods based on decomposition approaches are addressed.

A class of ring-opening polymerization processes are used for our industrial case studies. We develop rigorous dynamic reactor models for both semi-batch homopolymerization and copolymerization operations. The reactor models are based on first-principles such as mass and heat balances, reaction kinetics and vapor-liquid equilibria. We derive reactor models with both the population balance method and method of moments. The obtained reactor models are validated using historical plant data. Polymerization recipes are optimized with dynamic optimization algorithms to reduce polymerization times by modifying operating conditions such as the reactor temperature and monomer feed rates

over time. Next, we study scheduling methods that involve multiple process units and products. The resource task network scheduling model is reformulated to the state space form that offers a good platform for incorporating dynamic models. Lastly for the integration study, we investigate a process with two parallel polymerization reactors and downstream storage and purification units. The dynamic behaviors of the two reactors are coupled through shared cooling resources. We formulate the integration problem by combining the state space resource task network model with the moment reactor model. The case study results indicate promising improvements of process performances by applying dynamic optimization and scheduling optimization separately, and more importantly, the integration of the two.

# Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor Prof. Lorenz T. Biegler, who has been a tremendous mentor for me in academic and many other aspects. His wisdom, patience and enthusiasm towards research have greatly inspired and helped me in the past five years. It would be impossible for me to develop those invaluable knowledge and skills that give birth to this thesis work without him.

I would also like to express my sincere thanks to Dr. Carlos Villa and Dr. John Wassick from The Dow Chemical Company. I have greatly benefited from their strong support and generous help to my research project. The collaboration gives me a great learning experience outside the university. I have enjoyed working with them and their colleagues on various occasions.

My thanks go to the rest of my thesis committee members, Prof. Ignacio Grossmann, Prof. Shlomo Ta'asan, and Prof. Erik Ydstie, for their careful reading of my thesis and valuable comments during discussions.

I am grateful to the financial support from The Dow Chemical Company for my Ph.D. program as well as the internship opportunities in the past few summers.

I would also like to thank my friends here in the U.S. and back in China for their support and encouragement. I am in great debt to my parents for their unconditional love and support. Last, but never the least, I thank my beloved wife Qiwan. Qiwan has made my

life an enjoyable journey. Her love fills me with strength and courage to move forward, even in the most difficult times.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Scheduling and dynamic optimization are middle layers in the decision-making hierarchy of chemical processes, bridging between strategic and operational level decisions. Model based optimization methods for these decision-making processes have great potential to improve plant performance. Vertical integration of different layers in the hierarchy offers additional economic advantages. In this chapter, we present the background information and motivation of the integration problem for scheduling and dynamic optimization for chemical production processes.

## 1.1 Hierarchical Process Operations

The classic decision-making hierarchy consists of five decision layers as shown in Fig. 1.1: planning, scheduling, (dynamic) real time optimization, model predictive control, and regulatory control. The problem scope becomes narrower by traveling downward the pyramid, while the decision-making frequency increases. At the top, planning and scheduling are long-term decisions. Planning is used to create enterprise-wide production, distribution, sales and inventory plans based on customer and market information [1]. Schedul-

ing assigns limited resources (process units, materials, utilities, etc.) in a given production facility to create manufacturing sequences for products [2]. Decision frequency varies with respect to different processes. But typically, planning is performed yearly or monthly and scheduling is carried out monthly or weekly. The three layers at the bottom are related to operational decisions that focus on one/several specific process unit/units. Real time optimization (RTO) is used to update process operating conditions constantly to ensure successful execution of production plans made by top layers. RTO was first developed to deal with continuous processes using steady-state process models [3]. More recently, process dynamics have been directly incorporated into RTO problems to obtain solutions that give better economic benefit and operation flexibility [4]. The extended technology is termed as dynamic real time optimization (DRTO), which is able to optimize continuous processes with transit behaviors and batch processes, where operating steady-states do not exist by nature. Model Predictive Control (MPC) is a class of control algorithms that use explicit process models to update plant control inputs in real time. The objective of the MPC layer is to track the reference set-points/trajectories given by (D)RTO under process disturbances [5]. Regulatory control often refers to traditional PID controllers that directly interact with process units based on feedback mechanisms [6]. These controllers can be subsystems of an MPC structure.

## 1.2 Model Based Optimization of Process Operations

Increased energy cost, global competition, and stringent environment regulations are all driving forces for optimized decision-making in the chemical process industry. Model based optimization methods find their way into all layers of the decision hierarchy. These optimization approaches exploit equation based description of a process system, and translate an optimization task to a mathematical programming problem. The optimal decision is obtained as solving the corresponding math program to satisfy its optimal-

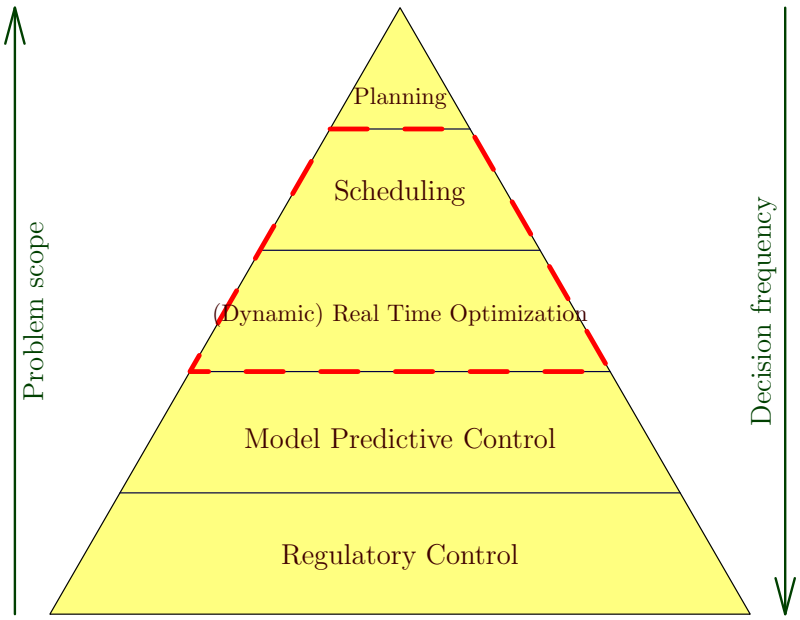



Figure 1.1: Decision-making hierarchy of chemical processes

ity condition. Model based optimization methods have been extensively studied in the academia, which become the core of the process systems engineering (PSE) discipline [7]. There is also a strong trend in the process industry to adopt these systematic problem solving tools to replace traditional methods that heavily rely on personal knowledge and expertise.

The process model structure and the type of algorithms used for optimization are different across different decision layers. Planning and scheduling involve many discrete decisions such as whether to built a new warehouse, which product to produce, how to sequence a group of products, etc. These give rise to discrete/binary variables in planning and scheduling problem formulations. In addition, these models often assume predetermined operating conditions such that process systems can be modeled with basic linear equations that roughly represent their behavior (e.g., mass balance, capacity limits). As a result, optimization of planning and scheduling is often posed as mixed-integer linear programs (MILPs). RTO employs detailed process models that explicitly consider the input-output relations of process units the interactions among them. These models can be

| | |
|---|---|
| Planning and scheduling | Shapiro [8], Grossmann [9], Floudas and Lin [10]<br>Méndez et al. [2], Harjunkoski et al. [11],<br>Maravelias and Sung [12], Verderame et al. [13] |
| RTO and DRTO | Marlin and Hrymak [3], Srinivasan et al. [14, 15]<br>Biegler [16], Kameswaran and Biegler [17] |
| MPC | García et al. [18], Morari and Lee [19]<br>Rawlings [20], Qin and Badgwell [5] |
| Regulatory control | Bequette [21] |

Table 1.1: Literature on model based optimization for decision-making

developed either from fundamental first-principles (conservation laws, thermodynamic relations) or data driven approaches (nonlinear regression). Both ways lead to continuous nonlinear algebraic models if the steady-state operation condition is assumed. The resulting continuous optimization problem can be solved by a number of nonlinear programming (NLP) algorithms. DRTO overcomes the limit of the steady-state assumption, and incorporates dynamic models that consist of differential algebraic equations (DAEs). This requires solving optimization problems with DAE constraints (a.k.a. dynamic optimization or optimal control problems). Dynamic optimization also applies to optimization problems in the MPC layer, as well as some of the regulatory control problems.

There is a vast body of literature on model based optimization approaches for chemical, petrochemical, and pharmaceutical processes. A number of well known references that summarize the developments and advances from the PSE community are listed in Tab. 1.1. These research studies make significant use of applied computational methods, especially math programming techniques, to model, analyze, and solve engineering problems that arise in the hierarchy.

## 1.3 Vertical Integration in the Decision-making Hierarchy

Conventionally, the decision-making process is one-directional, starting from the top to the bottom of the hierarchy. Therefore, decisions in each layer are optimized sequentially, and the complexity of the resulting optimization problems is well managed. However, two major disadvantages of this top-down decision-making approach are: first, degrees of freedom in lower-level decisions cannot be not fully explored by upper layer problems; second, top layer decisions cannot respond to disruptions that occur in the bottom. These lead to deteriorated plant performance in both profitability and reliability. A natural solution to that is to break the boundaries between the layers, and integrate the decision-making process. In fact, integration of production planning and scheduling has been investigated by many research groups over the past decade. The integrated formulations are posed as large-scale MILP problems that are often solved by decomposition algorithms [22, 23, 24, 25, 26]. Research studies on integrating RTO with process control are also reported [27, 28, 29]. Nonlinear model predictive control (NMPC) strategies are used to merge economic optimization and dynamic control functions into a unified dynamic optimization problem [30].

To finally integrate the strategic and operational decisions, work has to be done to tie process scheduling and DRTO technologies together (RTO can be viewed as simplified DRTO). An essential step towards this goal is to integrate process scheduling methods with dynamic optimization algorithms, which gives a combined optimization formulation that can design production schedules in conjunction with set-points/trajectories for process control. This is the main objective of the thesis study. A problem overview and literature review for integration of scheduling and dynamic optimization are discussed in the following section.

## 1.4 Integration of Scheduling and Dynamic Optimization

A single optimization formulation that can perform scheduling and dynamic optimization simultaneously is desirable to fulfill the task of integration. A number of major advantages of the integrated approach are noted by Engell and Harjunkoski [31]:

1. Improve production sequences such that reduce set-up and changeover costs, as well as lost during transitions.
2. Reduce maintenance needs and improve equipment life-time.
3. Improve the feasibility of schedules for operation.
4. Exploit the degrees of freedom of control in scheduling designs.
5. Use more precise and timely information in scheduling.

Similar perspectives are also given by Bassett et al. [32], Shobrys and White [33], and Harjunkoski et al. [34]. Although the integrated approach is favorable in terms of performance, its application has been rather limited. An integrated optimization formulation needs to deal with the discrete elements of scheduling and the dynamic characteristics of unit operation models. Moreover, the difference in time scale of scheduling and control needs to be handled. In terms of numerical optimization, this gives rise to mixed-integer dynamic optimization (MIDO) problems [35, 36] with multiple time periods. The computational complexity of such problems is quite high even for some small-size problems.

However, developments in optimization algorithms [37, 38] and computing power have enabled advances in this area over the last two decades. A number of successful research studies were first reported for continuous processes, especially commercial polymerization reactors. For example, Nyström et al. studied a grade sequencing and transition optimization problem of a polymerization process, where the transition trajectories, operating points, and sequencing of grades are determined all-at-once with a MIDO for-

mulation. Cost minimization is achieved by reducing raw material use and off-spec products [39]. Integrated optimization problems for continuous processes have been also studied by Flores-Tlacuahuac and Grossmann [40], Terrazas-Moreno et al. [41], Prata et al. [42], Busch et al. [43], Chu and You [44], and Zhuge and Ierapetritou [45]. Integration of batch processes was less often studied in the past, compared with continuous processes. Batch processes have more complicated scheduling environments, as assignments of processing tasks to process units are product-specific and time-dependent. The work by Bhatia and Biegler [46] was among the first to incorporate rigorous dynamic models into batch scheduling problems. Although the scheduling problem was restricted to flowshop plants with a special class of material transfer polices, this work suggested great potential of the integrated optimization approach to improve plant profitability in general. Recently, integration of batch processes starts to gain more attention. Mishra et al. [47] carried out a comparative study between two scheduling methods. The first was the standard approach with predetermined unit control recipes, while the other contained dynamic models of process units and recipes were generated together with schedules. The latter approach results in optimization problems with significantly larger sizes but higher overall profits. This indicates potential economic benefits in embedding rigorous process dynamic models into scheduling formulations, but also challenges in solving the resulting optimization problems. Research work along this line includes Romero et al. [48], Chu and You [49, 50], and Capón-García et al. [51]. These studies have adopted a top-down integration strategy, where the problem is viewed as enriching the higher level scheduling optimization formulation by replacing fixed task recipe parameters with detailed dynamic models. In a different vein, the integration problem can also be formulated from a bottom-up perspective, which is categorized as optimal control of hybrid discrete/continuous systems in control literature [52]. The guiding methodology is (hybrid) MPC algorithms . Task recipes are determined by optimizing the continuous control variables, while production schedules are represented by the discrete control variables in-

dicating task switch signals. Example studies are reported by Gallestey et al. [53] and de Prada et al. [54].

## 1.5 Research Statement and Thesis Outline

Integration of scheduling and dynamic optimization is an important research topic that is open to challenges in both problem formulation and model solution. This thesis work focuses on the development of such integrated optimization methods as well as their applications to industrial processes. The proposed integrated methods incorporate the recent advances in process scheduling and dynamic optimization, and extend them to explore the synergy between the two decision layers. It is desired to carry out a thoroughgoing research in this area, from methodology development to successful real-world implementations.

The remainder of the thesis is divided to two parts. In the first part, we review the optimization algorithms that are related to the solution of an integrated scheduling and dynamic optimization problem (we use the short note "the integrated problem" in later discussions for convenience). This part also includes a *proof-of-concept* study that aims at identifying the economic potential of the integration. In the second part, we apply and extend the optimization methods described in the first part to develop an integrate optimization strategy for an industrial polymerization process. The study covers topics such as reactor modeling, polymerization recipe optimization, process scheduling, and lastly integration.

# Part I

# Optimization Methodology

# Chapter 2

# Computational Methods

In this chapter, we present modeling and solution methods that are related to the integrated optimization problem. There is a great variety of methods developed for scheduling and dynamic optimization problems, respectively. We discuss these algorithms in comparison, including their advantages and disadvantages for the integrated problem. Also, we show available algorithms to solve integrated formulations.

## 2.1 Process Scheduling

Process scheduling is very diverse research area that has been studied by many different disciplines. Although heuristic/rule-based scheduling methods are efficient for certain problem types [55], we are interested in methods relying on mathematical programming that are generic and systematic. Optimization models for chemical production scheduling problems have very different structures, and systematic classification of them can be based on a number of criteria such as process type (batch and continuous), modeling of time (discrete and continuous), and process typology (sequential and network). For detailed description of scheduling model classification, refer to the review by Mar-

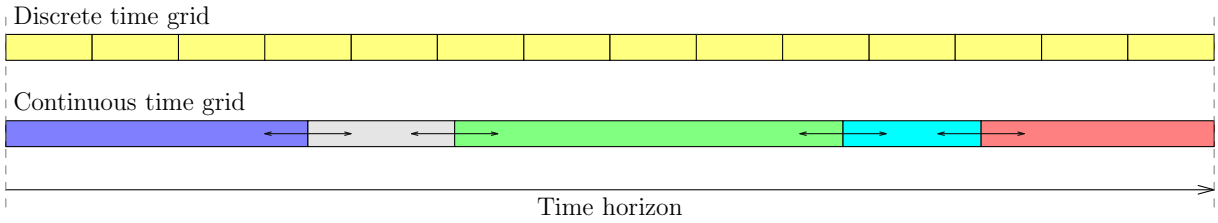



Figure 2.1: Discrete and continuous time representations

avelias [56]. In this section, we review a number of important concepts and solution approaches developed for scheduling optimization.

### 2.1.1 Time Representation

Time representation is a term that has been extensively used in PSE literature to refer to how time is modeled in a scheduling optimization formulation. Two well-known types are the discrete and continuous time representations. Fig. 2.1 is a simple illustration of the two modeling approaches. The discrete time approach divides the scheduling time horizon into a number of uniform time intervals of fixed lengths, and processing task lengths are integer multiples of the unit slot duration. The beginning and end of a discrete event (such as task start or end) must lie on the interval boundaries. In contrast, the continuous time approach introduces a moving time grid such that events can be placed at arbitrary time points. Usually, the continuous time representation requires fewer grid periods (time intervals/slots) compared with the discrete time approach.

Scheduling optimization formulations based on both time representations have discrete variables to model the state of a processing task in a time slot. For example, a binary variable is equal to one if a certain task starts at the beginning of a slot, and equal to zero otherwise. These discrete variables are often termed as *task assignment variables*. Continuous variables are also employed in scheduling models to represent continuous entries such as the (batch) size of a task. Additionally in continuous time models, continuous timing variables are incorporated to denote the lengths of time intervals.. The advan-

tages and disadvantages of the two time representations are compared in Tab. 2.1. The continuous and discrete time representations are applicable to the same model instances in general, but their computational performances are highly problem dependent.

Note that the continuous time representation has a number of variations. Global time grid models adopt a unique time grid for all process units [57, 58], while unit-specific time models introduce multiple asynchronized time grids for every unit [59, 60]. There are also scheduling models that focus on the precedence relationship among tasks rather than matching tasks with time intervals, for example, the general precedence models [61] and intermediate precedence models [62].

### 2.1.2 Process Representation

We are interested in scheduling problems with general network structures. Process representation methods are used as systematic tools to translate scheduling problems to math programming formulations. Two prevailing methodologies have been developed in the 90s, known as the state task network (STN) [63] and resource task network (RTN) [64] representations, respectively.

In STN models, states represent process materials including feeds, intermediates and final products and tasks are process operations that transform one/several input state(s) to output state(s). In a typical STN graph, state nodes are depicted with circles and task nodes with rectangles. The arcs that connect the two types of nodes represent possible interaction routes between states and tasks. The RTN framework extends the concept of *states* to *resources*. RTN models regard process equipment, material, and utility equivalently as resources that are consumed/generated by the execution of tasks. As a result, a RTN model has a more concise structure than its STN counterpart. Both the STN and RTN models are based on the discrete time representation initially, and continuous time models are developed in later studies [57, 65].

| | |
|---|---|
| Model accuracy | Continuous: high, continuous timing variables<br>Discrete: low, round-up errors in task lengths |
| Model size | Continuous: small, sparse time grids<br>Discrete: large, dense time grids |
| Formulation tightness | Continuous: low, big M terms in timing constraint<br>Discrete: high, low integrality gap |
| Consistency | Continuous: low, customization for different applications<br>Discrete: high, concise and consistent structure |
| Time-dependent event | Continuous: indirect, additional constraints needed<br>Discrete: direct, known time grid point positions |

Table 2.1: Pros and cons for the two time representation approaches

Alternative process representation approaches for scheduling problems have also been studied in the PSE community, such as the state sequence network [66]. In addition, the state equipment network (SEN) [67, 68] has been applied in process synthesis and design problems that share similarities with scheduling problems (a unit switches between different operating modes). SEN models employ an equipment-oriented modeling perspective, focusing on the connectivity between process units rather than tasks. One advantage of SEN models is that tasks performed by the same unit can be described in a compact and consistent way, which is favorable for problems with rigorous unit models. As a result, the SEN method can be carried over to integrated scheduling and dynamic optimization problems, where process units are described with detailed dynamic models. We discuss the SEN based integration in the next chapter.

### 2.1.3 Solution Algorithms for Scheduling Models

Most of the process scheduling models are formulated as MILPs, which can be written in a generic way as:

$$\begin{aligned} \min \quad & a^{\mathrm{T}}x + b^{\mathrm{T}}w \\ s.t. \quad & Ax + Bw \leqslant d, \\ & x \in \mathbb{R}^{n_x}, w \in \{0,1\}^{n_w}. \end{aligned} \tag{2.1}$$

Equation (2.1) uses binary discrete variables $w$ and continuous variables $x$. MILP problems are NP-complete in terms of computational complexity [69], thus the problem complexity grows dramatically with an increasing number of discrete variables. Two major classes of algorithms for solving MILPs are the branch and bound method and cutting plane approach [70]. The branch and bound method adopts a *divide-and-conquer* strategy. The original MILP problem is solved through a sequence of restricted problems that comprises a tree structure. The tree is constructed by branching on the elements of discrete variables $w$ recursively. At each node of the branch and bound tree, a linear program (LP) relaxation of the original MILP problem is solved (dropping the integrality requirement on $w$). If a LP solution happens to have integral $w$ values, it is recorded as a candidate solution giving the upper bound of the MILP problem. Meanwhile, the lower bound is obtained via solving the root note relaxation, and updated through the branching procedure. The algorithm converges when the best upper bound coincides with the current lower bound. The cutting plane methods are based on the valid inequalities that recursively refine the feasible set of a MILP by eliminating regions that are only feasible for its LP relaxations. Accumulation of these valid inequalities allows the relaxed solution to approach the optimal solution of the MILP problem. Well known cutting plane methods include Gomory's cut [71], *lift-and-project* cut [72], etc. Modern MILP algorithms often combine the cutting planes with branch and bound procedures, which give the *branch-and-cut* methods [73]. Current commercial MILP solvers mostly implement the *branch-*

*and-cut* strategy such as CPLEX [74] and Gurobi [75].

## 2.2 Dynamic Optimization

Dynamic optimization finds many applications in chemical processes, including *off-line* problems such as batch recipe optimization and transition operation optimization, as well as *on-line* implementations such as predictive control [76]. Usually the embedded DAEs in a dynamic optimization problem are couched in the continuous time domain. Equation (2.2) gives a general form of dynamic optimization problems:

$$\begin{aligned}
\min_{u(t),p} \quad & \Phi(z(t_f)) \\
s.t. \quad & \dot{z}(t) = f(z(t), y(t), u(t), p), \quad z(t_0) = z_0, \\
& g(z(t), y(t), u(t), p) = 0, \\
& h_e(z(t), y(t), u(t), p) = 0, \\
& h_i(z(t), y(t), u(t), p) \leqslant 0, \\
& z \in \mathbb{R}^{n_z}, y \in \mathbb{R}^{n_y}, u \in \mathbb{R}^{n_u}, p \in \mathbb{R}^{n_p}.
\end{aligned} \tag{2.2}$$

Here, the problem is defined from time $t_0$ to $t_f$. For variable notation, $z$ and $y$ are the differential and algebraic state variables, respectively, $u$ represents the control variables, and $p$ is time independent decision variables. The objective function $\Phi(\cdot)$ is written in the Mayer form as an end-cost function of differential states. In process optimization problems, the functions $f(\cdot)$ and $g(\cdot)$ note the DAE system that represents process models, which can be either first-principles or data-driven. The initial condition of the differential states is known as parameter $z_0$. It is also possible that initial conditions are replaced by boundary conditions in certain circumstances. The equalities $h_e(\cdot)$ and inequalities $h_i(\cdot)$ denote the optimization constraints that come from process safety regulations, product quality requirements, etc. Note that the bounds on the state and control variables are also

included in $h_i(\cdot)$. The constraints can be enforced either at final time by restricting $t = t_f$ (end-point constraints) or within the whole time horizon $t \in [t_0, t_f]$ (path constraints).

### 2.2.1 Solution Approaches for Dynamic Optimization

Analytical solution of the dynamic optimization problem requires applying Pontryagin's maximum principle or solving Hamilton-Jacobi-Bellman equations, which are quite difficult for realistic problems and they cannot handle inequality constraints. Therefore, numerical solution techniques are favored to obtain approximated optimal solutions. Numerical solution methods for dynamic optimization problems are generally separated into two classes, known as the *sequential* [77, 78, 79, 80] and *simultaneous* [81, 82, 83] approaches. In both methods, the decision variable (control policy) is parameterized by using an appropriate function approximation (e.g., piecewise constant parameterization). This procedure is often called as *control vector parameterization* in literature. The difference between the two classes is due to the treatment of the embedding DAEs.

#### Sequential Approach

The sequential approach formulates a reduced size NLP that treats the DAE system as an extrinsic *black box*, where the state trajectories and sensitivities required in optimization search are provided by external numerical integration packages for DAEs. The sensitivities are derivative information of the DAE system with respect to the decision variables, and the sensitivity equations are also a group of DAEs that can be solved simultaneously with the original system. Two types of sensitivities are often used. Direct sensitivities [84] are defined with respect to state variables, and adjoint sensitivities [85] are defined over objective and constraint equations. A schematic of the sequential method (also termed as single shooting) is depicted in Fig. 2.2. The sequential method relies on repeated integration of DAEs so that optimization is subject to failure for unstable systems, where DAE

integrators cannot converge. Moreover, path constraints and inequality constraints cannot be directly handled in sensitivity analysis. Reformulation is needed to convert them to either end-point constraints or penalty terms in the objective function.

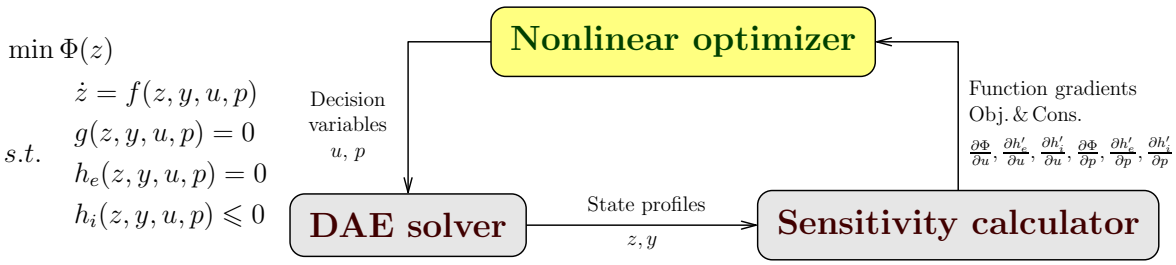



Figure 2.2: Schematic diagram of the sequential method

### Simultaneous Approach

The simultaneous approaches include *multiple shooting* [81] and *simultaneous collocation* [82] methods. In multiple shooting, the time horizon of interest is divided into several segments, and the sequential approach is applied to each time slot. This method provides potential of handling unstable systems while increasing the size of resulting problems. The simultaneous collocation approach (or direct transcription) is motivated by avoiding explicit integration of DAEs that is required by the two shooting approaches. The state variables are parameterized via orthogonal collocation over finite elements (OCFE) [86]. In the collocation method, the time domain is divided into a finite number of finite elements, and several collocation points are placed in each element. Orthogonal polynomials are introduced at collocation points, and the state profiles are approximated by a weighted summation of the orthogonal polynomials (illustrated in Fig. 2.3). The approach corresponds to a particular implicit *Runge-Kutta* method with high order accuracy and excellent stability properties [16].

The collocation equation for differential state variables in a finite element can be stated as

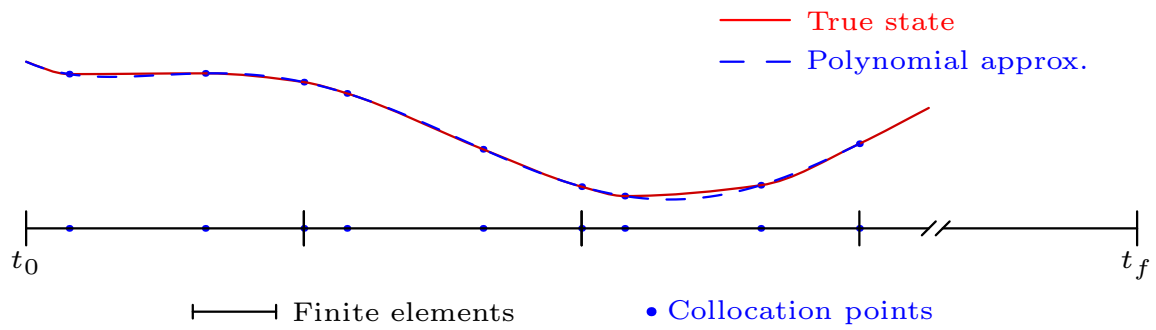



Figure 2.3: Illustrative example of OCFE

below, if the Runge-Kutta basis is used:

$$z(t) = z_j^0 + h_j \sum_{k=1}^{K} \Omega_k(\tau) \dot{z}_{j,k}, \quad j = 1, ..., J. \tag{2.3}$$

Here, $j$ is the index of finite elements from 1 up to $J$ and $k$ is the index for collocation points from 1 to $K$, $h_j$ is the length of finite element $j$, $\tau \in [0, 1]$ is the normalized time coordinate within an element and $t = t_j^0 + h_j \tau$ ($t_j^0$ denotes the starting time of element $j$), $z_j^0$ is the value of the differential state at the beginning of element $j$, $\dot{z}_{j,k}$ is the value of the first derivatives at collocation points, and $\Omega_k$ is a polynomial of $\tau$ of order $K$, defined as

$$\Omega_k = \int_0^\tau \ell_k(\tau') \mathrm{d}\tau', \quad k = 1, ..., K. \tag{2.4}$$

Here, $\ell_k$ is an orthogonal basis function. Lagrange interpolation polynomials are often favored owing to their exactness at collocation points, which are shown as:

$$\ell_k(\tau) = \prod_{k'=1, \neq k}^{K} \frac{\tau - \tau_{k'}}{\tau_k - \tau_{k'}}, \quad k = 1, ..., K. \tag{2.5}$$

In addition, the continuity condition across finite element boundaries is preserved:

$$z_{j+1}^0 = z_j^0 + h_j \sum_{k=1}^{K} \Omega_k(1) \dot{z}_{j,k}, \quad j = 1, ..., J - 1. \tag{2.6}$$

The algebraic states $y$ can be treated similarly with the orthogonal polynomials:

$$y(t) = \sum_{k=1}^{K} \ell_k(\tau) y_{j,k}, \quad j = 1, ..., J. \tag{2.7}$$

However, the continuity condition is not necessarily enforced for algebraic states. The equations of the DAE system and process constraints (constraints in formulation (2.2)) are only enforced at collocation points, and the equation form remains the same. By this means, the dynamic optimization formulation (2.2) is translated to a NLP problem:

$$\begin{aligned}
\min_{u(t),p} \quad & \Phi(z_{j,k}) \\
s.t. \quad & z_{j,k} = z_j^0 + h_j \sum_{k=1}^{K} \Omega_k(\tau_k) \dot{z}_{j,k}, \\
& z_{j+1}^0 = z_j^0 + h_j \sum_{k=1}^{K} \Omega_k(1) \dot{z}_{j,k}, \quad z_1^0 = z_0, \\
& \dot{z}_{j,k} = f(z_{j,k}, y_{j,k}, u_{j,k}, p), \\
& g(z_{j,k}, y_{j,k}, u_{j,k}, p) = 0, \\
& h_e(z_{j,k}, y_{j,k}, u_{j,k}, p) = 0, \\
& h_i(z_{j,k}, y_{j,k}, u_{j,k}, p) \leqslant 0.
\end{aligned} \tag{2.8}$$

The simultaneous collocation method has a number of significant advantages for process optimization problems. First, it is able to handle dynamic systems with unstable modes. The optimality condition of the dynamic optimization formulation (2.2) can be cast into a boundary value problem (BVP) [87], and the optimality condition of the translated NLP (2.8) ((known as Karush-Kuhn-Tucker (KKT) condition [88])) is equivalent to applying collocation on the BVP. The NLP benefits from the *dichotomy property* of BVPs that can *pin down* the unstable modes [89, 90]. Meanwhile, NLP problems after collocation have sparse structures that can be efficiently exploit by NLP algorithms, which offers an edge particularly for *on-line* applications [91]. Last but not least, process constraints

can be treated directly within the discretized scheme without the burden of reformulation.

### 2.2.2 Nonlinear Programming Methods

All dynamic optimization approaches rely on NLP solvers to conduct optimization searches on the decision variables. Newton type solvers are generally preferred due to their fast convergence properties [76, 92]. A number of popular NLP algorithms are listed in Tab. 2.2, together with available code implementations. The sequential quadratic programming (SQP) method solves a sequence of quadratic programs (QP) to guide the search. The QPs are constructed by using the variable value and derivative information at each iteration. SQP solvers are favored by the sequential method for dynamic optimization. First, the NLP problem size is relatively small in the sequential approach; also, second order derivatives are very expensive to compute via sensitivity analysis, but SQP methods generally perform well with approximated derivatives. The generalized reduced gradient (GRG) method partitions the variables in a NLP into basic, nonbasic, and superbasic variables. The basic variables are used to solve equality constraints, nonbasic variables are fixed at either their upper or lower bounds, and superbasic variables are used to drive the optimization search. The interior point method reformulates the inequality constraints as barrier (penalty) terms to the objective function, and solves a series of NLP problems with decreasing barrier parameters to recover the optimal solution of the original problem. The method has advantages dealing with large-scale problems with many degrees of freedom. Therefore, it is often used to solve the NLP problems resulting from the simultaneous collocation method.

| Algorithm class | Solver and reference |
|---|---|
| Sequential quadratic programming | SNOPT [93], filterSQP [94, 95] |
| Generalized reduced gradient | CONOPT [96, 97], MINOS [98] |
| Interior point method | IPOPT [99], KINTRO [100] |

* Listed solvers are examples for each algorithm class (not exhaustive).

Table 2.2: NLP algorithms and solvers

## 2.3 Integrated Scheduling and Dynamic Optimization

Integration of scheduling and dynamic optimization results in optimization problems with both discrete decision variables and differential constraint equations, which can be formulated as MIDO problems. Given the combined complexity of discrete decision variables and differential equation constraints, MIDOs cannot be solved directly, and most solution strategies are designed to tackle the discrete and dynamic components in a separate manner. *Reformulation* is needed to convert MIDOs to a form that can be handled by existing optimization solvers. The two viable approaches for dynamic optimization, the sequential and simultaneous methods, can carry out this task. Barton and coworkers have proposed a series of approaches that apply the sequential method to MIDOs [35, 101, 102]. In this manner, a MIDO problem is decomposed similarly as suggest in Fig. 2.2, except for the optimization search involves MINLP algorithms instead of NLPs. The authors have also investigated global optimization methods for MIDOs that rely on convex function relaxations [103, 104]. On the other hand, the simultaneous collocation method has been adopted in many recent research studies dealing with MIDOs [41, 51, 105]. This approach translates a MIDO problem to a purely algebraic MINLP that is nonconvex. In addition, a list of MINLP algorithms and solvers is given in Tab. 2.3. These algorithms guarantee global optimality for convex MINLPs, and can be applied to find good feasible solutions for nonconvex ones. Decomposition methods for MIDOs have also been studied, for example, by Terrazas-Moreno et al. [115] and Bansal et al. [36]. The main idea for these decomposition methods is to separate the discrete decisions and dynamic process

| Algorithm class | Solver | | | |
|---|---|---|---|---|
| | SBB [106] | DICOPT [107] | BONMIN [108] | Alpha-ECP [109] |
| NLP-based branch and bound [110] | × | | × | |
| Generalized Benders decomposition [111] | | | × | |
| Outer Approximation [112] | | × | × | |
| Extended cutting plane [113] | | | | × |
| LP/NLP-based branch and bound [114] | | | | |

* Listed solvers are examples for each algorithm class (not exhaustive).

Table 2.3: MINLP algorithms and solvers

models, while exploring the synergy between the two by solving the separated problems in certain iterative procedures. In Chapter 7, we discuss a solution method based the generalized Benders decomposition. Lastly, it is also worth noting that a MIDO formulation can be equivalently stated as a mixed-logic dynamic optimization problem [116] by replacing the discrete variables with logical disjunctions [117]. This logic based representation method is used in our *proof-of-concept* study for integration, which is presented in the following chapter.

# Chapter 3

# Integration Using State Equipment Networks

This chapter presents a *proof-of-concept* study for the integration of process scheduling and dynamic optimization to identify its potential in improving the economic performance of chemical batch processes. The state equipment network (SEN) is used to represent a process system. Modeling based on the SEN framework invokes both logical disjunctions and operational dynamics; thus the integrated formulation leads to a mixed-logic dynamic optimization (MLDO) problem. The MLDO problem is translated into a MINLP for model solution, using the Big M reformulation and the simultaneous collocation method. Two case studies are demonstrated to show the advantages of the integrated approach over the conventional recipe-based scheduling method.

## 3.1 Problem Statement

The integrated approach aims at improving the overall performance of a batch process by simultaneously optimizing its production schedule and corresponding control profiles of

the process units involved. The problem can be stated as follows:

Given:

*Plant configuration*

⋆ a set of process units and their uses

⋆ feed materials, intermediates and final products and their constituents

⋆ information on production sequences or steps

*Scheduling preknowledge*

⋆ a fixed scheduling time horizon (except for makespan problems)

⋆ target performance index, e.g., maximum profitability, minimum makespan or earliness

⋆ transfer policy for each material

⋆ procurement price of raw materials

⋆ market price, demand and quality requirements of final products

⋆ market price of utilities consumed in batch operations

*Process dynamics*

⋆ dynamic (or simplified) models of process operations

⋆ bounds on state and control variables corresponding to possible process operations

⋆ additional constraints regarding safety and quality issues

Determine:

⋆ optimal production sequences and timings of all batch units

⋆ optimal control profiles of batch units

⋆ optimal performance index value

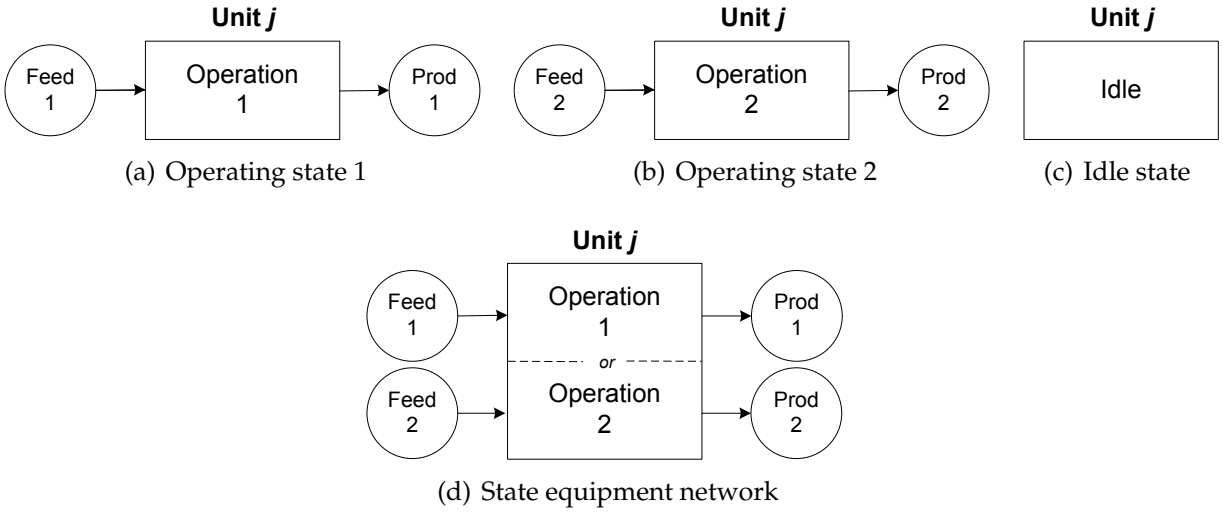
A desired comprehensive scheduling formulation is expected to handle many aspects of manufacturing considerations: order fulfillment, product transitions and shared resources, to name a few. Albeit simplified, we will restrict our problem to a more concise status, where the objective is to maximize the net profit at the end of scheduling horizon with no change-over delays and fulfillment deadlines, while the unlimited intermediate

storage (UIS) policy is assumed. Nonetheless, extensions can be carried out within the SEN infrastructure, using similar techniques developed for the STN and the RTN [118].

## 3.2 The State Equipment Network

The original SEN framework for process synthesis and design [67, 119] adopts a directed graph with two basic types of nodes to represent a batch process. These nodes are namely, the *state* nodes, including all sorts of materials involved in the process, and the *equipment* nodes, containing all processing units in the plant. However, a striking feature of batch plants is their capability of flexible manufacturing, which invokes discrete time-varying behavior of process units, such as switching between different operations and turning on and off. As a result, the mathematical description of an equipment node is subject to change over time as well as the connectivity of a SEN graph. Therefore, an additional temporal dimension of combinatorial decisions has to be incorporated into the original SEN in order to designate the status of equipment nodes over time. As equipment and material state nodes are connected by directed edges alternatively, a scheduling problem can be reinterpreted as routing material flows through the network. Last but not least, since a batch unit may accommodate multiple operations, the term *state* here can also be defined as an operational status of the unit. To avoid ambiguity, we will use the term *operating state* to indicate *operation* and the term *material* to refer to *material state*.

To better interpret how the SEN tackles a batch scheduling problem, a simple instance of a single-stage batch plant is depicted in Fig. 3.1. The unit serves for two different manufacturing purposes. It converts the corresponding feed material to the product, if switched to a particular operating state. This plant has three possible setups, shown in Figs. 3.1(a)-3.1(c). The SEN representation drawn in Fig. 3.1(d) is able to capture all these three potential configurations of the system and postulate them together with disjunctive




(a) Operating state 1 (b) Operating state 2 (c) Idle state

(d) State equipment network

Figure 3.1: An illustrative example of applying the state equipment network

propositions (the idle state is included implicitly).

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

### 3.2.1 Scheduling Using the SEN

To better suit the equipment control perspective, the scheduling model adopts the unit-specific event-based continuous-time representation [10], where a scheduling horizon is divided up into a finite number of event slots for each unit. These slots can be asynchronous from one unit to another, thus the scheduling formulation asks for a group of sequencing constraints to adjust the global sequencing behavior among different pieces

of equipment. Moreover, for each process unit, its operating states in event slots are labeled with a series of binary variables. The remainder of this section will take the reader through the five major aspects of scheduling considerations in the SEN. Variable notation used in this chapter is listed in Sec. 3.5 (page 51).

### Assignment Constraints

Assignment constraints allocate operating states in both units and time horizon:

$$\sum_{s \in \mathcal{S}_j} w_{j,s,t} \leqslant 1 \quad \forall j \in \mathcal{J}, t \in \mathcal{T}. \tag{3.1}$$

In Eq. (3.1), binary variable $w_{j,s,t}$ indicates unit $j$ is in operating state $s$ in current time slot $t$, if equal to $1$. In the SEN representation, operating states of the same unit are exclusive of each other because equipment cannot be shared. In addition, it is also possible that the unit is not occupied by any operating states during certain periods of time such that all binaries are equal to $0$. In short, the assignment of operating states allows at most one operating state to be active in a unit per event slot.

### Material Balance

An event point is defined as the time point where an event slot starts ($t$ is used as the index for both the slot and point). At such a point, balance equations are enforced for every material $r$ to keep track of its available amount, before moving into the next time interval. Moreover, at event point $t$, materials that are not used by any process operation

executed in event slot $t$ are termed as excess materials $R_{r,t}$.

$$R_{r,t} = R_{r,t-1} + \sum_{j \in \mathcal{J}_r^p} R_{j,r,t-1}^p - \sum_{j \in \mathcal{J}_r^c} R_{j,r,t}^c, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}, t > 1; \tag{3.2a}$$

$$R_{r,1} = R_{r,0} - \sum_{j \in \mathcal{J}_r^c} R_{j,r,1}^c, \quad \forall r \in \mathcal{R}; \tag{3.2b}$$

$$R_{r,0} = \overline{R}_r, \quad \forall r \in \mathcal{R}' \subseteq \mathcal{R}. \tag{3.2c}$$

Equation (3.2a) states that the amount of excess material $r$ at event point $t$ equals its value at the previous event point $t-1$ adjusted by the amount consumed within slot $t$, and produced within slot $t-1$; however, at the first event point, no production has been obtained yet (Eq. (3.2b)). A process material can be generated or consumed by one or several units, denoted as subsets $\mathcal{J}_r^p$ and $\mathcal{J}_r^c$, respectively. The initial amount of material $r$ ($R_{r,0}$) may be specified accordingly as stated in Eq. (3.2c). Usually, the initial amounts of intermediate and final materials are assigned to the given parameters $\overline{R}_r$, while the amounts of raw materials of procurement cost remain variables, i.e., in Eq. (3.2c) set $\mathcal{R}' = \mathcal{R} \setminus \mathcal{R}_{raw}$.

### Capacity Constraints

Typically, the extent of a batch operation is limited by the vessel size of the equipment used and also pertinent process safety considerations. Equation (3.3) helps enforce these restrictions on the batch size $b_{j,t}$.

$$\sum_{s \in \mathcal{S}_j} w_{j,s,t} B_j^{min} \leqslant b_{j,t} \leqslant \sum_{s \in \mathcal{S}_j} w_{j,s,t} B_j^{max}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}. \tag{3.3}$$

Meanwhile, the amount of excess material $R_r$ must stay within a certain range that is stipulated by regulations such as safety stock, storage limit and so forth (Eq. (3.4)). Like other unit-specific formulations, $R_r$ does not necessarily agree with the material inventory

in real time, since event points of units are most likely asynchronized. In this formulation, we avoid this problem by assuming the UIS policy. Nevertheless, rigorous treatment of other operational philosophies can be amended at the cost of additional variables and constraints [120].

$$R_r^{min} \leqslant R_{r,t} \leqslant R_r^{max}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}. \tag{3.4}$$

**Timing Constraints**

In this continuous-time formulation, the $t^{\text{th}}$ event point of batch unit $j$ is denoted as $T_{j,t}$ and the associated processing time of the unit is called $Tp_{j,t}$. Since the number of these timing variables is proportional to the number of batch units, the SEN generally leads to smaller size models compared to the conventional task-oriented formulations. To sequence operations, Eq. (3.5) is first applied to individual units. Here, no overlapping is allowed for any neighboring pair of event slots of a unit, i.e., the starting time of an event slot cannot be earlier than the time when its precedent slot ends.

$$T_{j,t+1} \geqslant T_{j,t} + Tp_{j,t}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}, t < |\mathcal{T}|\,. \tag{3.5}$$

Besides, to obtain appropriate alignments of event slots of different units that use the same intermediate material, a second group of constraints is introduced. For instance, if material $r$ is produced by unit $j$ at the end of time slot $t$, and another unit $j'$ has active operating state $s'$ in event slot $t'$ ($t < t'$) that consumes the material, then event point $t'$ of unit $j'$ should be placed after the end of slot $t$ of unit $j$ in real time.

$$\begin{aligned} &T_{j',t'} \geqslant T_{j,t} + Tp_{j,t} - H(2 - \sum_{s \in \mathcal{S}_j \cap \mathcal{S}_r^p} w_{j,s,t} - \sum_{s' \in \mathcal{S}_{j'} \cap \mathcal{S}_r^c} w_{j',s',t'}), \\ &\forall r \in \mathcal{R}, j \in \mathcal{J}_r^p, j' \in \mathcal{J}_r^c, j \neq j', t, t' \in \mathcal{T}, t < t' \leqslant t < |\mathcal{T}| \end{aligned} \tag{3.6}$$

This constraint is only enforced when both the production and consumption actions take place, thus it is relaxed by a big M parameter $H$, which is the given scheduling horizon. Also, all processing operations should be completed within the specified time horizon for a valid schedule. For this reason, a group of bounding constraints are introduced as Eq. (3.7).

$$T_{j,t} \leqslant H, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}. \tag{3.7a}$$

$$T_{j,t} + Tp_{j,t} \leqslant H, \quad \forall j \in \mathcal{J}, t = |\mathcal{T}|. \tag{3.7b}$$

**Quality Measurement**

This set of variables and constraints does not appear in conventional batch scheduling models, due to the assumption of well-executed recipes. Under this assumption, no quality giveaways will occur after running a batch for multiple times, and therefore there is no need to retain quality information. However, when operational strategy varies in time, it consequently results in variations in material quality, such as concentration distribution in a mixture. To account for this in the integrated scheme, we define $\eta_{r,c,t}$ as the fraction of component $c$ in excess compound material $r$ (noted as $\mathcal{C}_r$) at event point $t$, as well as its counterpart $\phi_{j,r,c,t}$ for material $r$ produced by unit $j$ at the end of event slot $t$. Material flows coming from different sources are likely to have nonidentical composition distributions. A way to deal with this is to assume a *blending-before-using* regulation, that is, newly produced materials are fully blended with inventories before they are used in the next processing stage. Under this circumstance, an excess mixture after blending has the averaged composition of all source flows, and the quality of the materials is measured by the mean value as shown in Eq. (3.8).

$$\eta_{r,c,t} = (\eta_{r,c,t-1} R_{r,t-1} + \sum_{j \in \mathcal{J}_r^p} \phi_{j,r,c,t-1} R_{j,r,t-1}^p)/(R_{r,t-1} + \sum_{j \in \mathcal{J}_r^p} R_{j,r,t-1}^p),$$
$$\forall r \in \mathcal{R}, c \in \mathcal{C}_r, t \in \mathcal{T}, t > 1. \tag{3.8}$$

On the other hand, in order to ensure specific requirements of material quality, especially for final products, additional constraints are enforced to restrict $\eta_{r,c,n}$. These constraints can be generally written as Eq. (3.9).

$$H(\eta_{r,c,t}, \overline{\eta}_{r,c}) \geqslant 0, \quad \forall r \in \mathcal{R}, c \in \mathcal{C}_r, t \in \mathcal{T}. \tag{3.9}$$

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

### 3.2.2 Dynamic Optimization in the SEN

The SEN-based framework offers a great opportunity to model hybrid discrete/continuous systems. For an operation, the dynamic model is couched in continuous time with a finite time length. In terms of process scheduling, the discrete decisions are represented via discrete events taking place at distinct time points. Logical disjunctions are applied to tie the discrete and the continuous parts together.

**Dynamic Models of Operating States**

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

$$\begin{aligned}
\tfrac{dz_{j,t}(\tau)}{d\tau} &= f_{j,s}(z_{j,t}(\tau), y_{j,t}(\tau), u_{j,t}(\tau))Tp_{j,t} \\
g_{j,s}(z_{j,t}(\tau), y_{j,t}(\tau), u_{j,t}(\tau)) &= 0 \\
z_{j,s}^{min} \leqslant z_{j,t}(\tau) &\leqslant z_{j,s}^{max} \\
y_{j,s}^{min} \leqslant y_{j,t}(\tau) &\leqslant y_{j,s}^{max} \\
u_{j,s}^{min} \leqslant u_{j,t}(\tau) &\leqslant u_{j,s}^{max} \\
Tp_{j,s}^{min} \leqslant Tp_{j,t}(\tau) &\leqslant Tp_{j,s}^{max}
\end{aligned} \qquad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, t \in \mathcal{T} \tag{3.10}$$

In Eq. (3.10), all the variables are defined over unit $j$. However for different operating states of the unit, the dynamic models are constructed in dissimilar manners. Furthermore, the dynamic DAEs used here are preferred to be of index one and reformulation is recommended for higher index systems, so as to guarantee solution uniqueness and numerical robustness. In addition to Eq. (3.10), the initial condition of the differential state variable $z_{j,n}$ must be specified. Both the batch size and the inlet material quality are considered as the input variables to the dynamic system, given in Eq. (3.11).

$$z_{j,t}(0) = Z_{j,s}(b_{j,t}, \eta_{r,c,t}), \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, r \in \mathcal{R}_s^c, c \in \mathcal{C}_r, t \in \mathcal{T}. \tag{3.11}$$

Material production and consumption are important variables that link the intrinsic dynamic system of a batch unit to external plant environment. Often in practice, input

materials are considered to be consumed at the beginning ($\tau = 0$) while output products become available when an operation ends ($\tau = 1$). And over the time interval, the batch unit is occupied. Here, material inputs and outputs are described via functions of state variables as shown in Eq. (3.12). However, for semi-batch operations where feed materials are continuously transferred into a unit, the consumption functions can be integrals over the residence time.

$$R^p_{j,r,t} = R^p_{j,s}(z_{j,t}(1), y_{j,t}(1)), \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, r \in \mathcal{R}^p_s, t \in \mathcal{T}; \tag{3.12a}$$

$$R^c_{j,r,t} = R^c_{j,s}(z_{j,t}(0), y_{j,t}(0)), \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, r \in \mathcal{R}^c_s, t \in \mathcal{T}. \tag{3.12b}$$

Likewise, product quality can be written as functions of state variables as well. Moreover, a quality measurement function may be path dependent, i.e., in the form of an integral over time $\tau = [0, 1]$. In general, Eq. (3.13) is able to represent product quality with proper definitions of state variables.

$$\phi_{j,r,c,t} = \phi_{j,s}(z_{j,t}(\tau), y_{j,t}(\tau)), \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, r \in \mathcal{R}^p_s, c \in \mathcal{C}_r, t \in \mathcal{T}, \tau \in [0, 1]. \tag{3.13}$$

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

$$\Phi_{j,s}(\phi_{j,r,c,t}, \overline{\phi}_{j,r,c}) \leqslant 0, \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, r \in \mathcal{R}^p_s, c \in \mathcal{C}_r, t \in \mathcal{T}. \tag{3.14}$$

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

active operating state, process control inputs, execution times and batch size.

$$F_{j,t} = F_{j,s}(u_{j,t}(\tau), Tp_{j,t}, b_{j,t}), \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j, t \in \mathcal{T}, \tau \in [0,1]. \tag{3.15}$$

**Disjunctions of Batch Units**

As we described earlier, in contrast to task-oriented networks, the SEN regards different operating states as individual dynamic systems inside disjuncts but with a consistent variable definition. Therefore, the disjunction for a unit comprises the disjuncts of the operating states that the unit can perform. Besides, one additional disjunct is added to account for equipment idling. In the idle state, no dynamic equations are needed to describe the behavior of the unit, and all of the interfacing variables such as material production and consumption are fixed to zero. The modeling approach is usually addressed as generalized disjunctive programming (GDP) [121]. It recasts discrete optimization problems into logic-based models, offering structural advantages both in model formulation and solution [122]. In the context of the SEN, a disjunction captures the essence of exclusive choices between operating states with the assistance of assignment constraints. With this approach, dynamic models of operating states are organized by disjunctions as Eq. (3.16) states.

$$\begin{bmatrix} w_{j,s,t} = 1 \\ \frac{dz_{j,t}(\tau)}{d\tau} = f_{j,s}(z_{j,t}(\tau), y_{j,t}(\tau), u_{j,t}(\tau))Tp_{j,t} \\ g_{j,s}(z_{j,t}(\tau), y_{j,t}(\tau), u_{j,t}(\tau)) = 0 \\ z_{j,s}^{min} \leqslant z_{j,t}(\tau) \leqslant z_{j,s}^{max} \\ y_{j,s}^{min} \leqslant y_{j,t}(\tau) \leqslant y_{j,s}^{max} \\ u_{j,s}^{min} \leqslant u_{j,t}(\tau) \leqslant u_{j,s}^{max} \\ Tp_{j,s}^{min} \leqslant Tp_{j,t}(\tau) \leqslant Tp_{j,s}^{max} \\ z_{j,t}(0) = Z_{j,s}(b_{j,t}, \eta_{r,c,t}) \\ R_{j,r,t}^{p} = R_{j,s}^{p}(z_{j,t}(1), y_{j,t}(1)) \\ R_{j,r,t}^{c} = R_{j,s}^{c}(z_{j,t}(0), y_{j,t}(0)) \\ \phi_{j,r,c,t} = \phi_{j,s}(z_{j,t}(\tau), y_{j,t}(\tau)) \\ \Phi_{j,s}(\phi_{j,r,c,t}, \overline{\phi}_{j,r,c}) \leqslant 0 \\ F_{j,t} = F_{j,s}(u_{j,t}(\tau), Tp_{j,t}, b_{j,t}) \end{bmatrix}_{s \in \mathcal{S}_j} \bigvee \begin{bmatrix} \sum_{s \in \mathcal{S}_j} w_{j,s,t} = 0 \\ R_{j,r,t}^{p} = 0 \\ R_{j,r,t}^{c} = 0 \\ Tp_{j,t} = 0 \\ F_{j,t} = 0 \end{bmatrix} \quad \forall j \in \mathcal{J}, t \in \mathcal{T}, \tau \in [0,1]. \tag{3.16}$$

### 3.2.3 Integration with SEN

In the integrated formulation, the objective function is to maximize the revenue for a given scheduling horizon, which is equal to product sales minus raw materials and operating costs. The total amount of products are counted at the end of the last event slot. Meanwhile feedstock is purchased in advance and cost of unit operations is accumulated along the time horizon. The constraints of the integrated formulation come from both the

scheduling and unit operation respects. In sum, the problem can be stated as follows:

$$\max \quad \sum_{r\in\mathcal{R}_{prod}} P_r(R_{r,T} + \sum_{j\in\mathcal{J}_r^p} R_{j,r,T}^p) - \sum_{r\in\mathcal{R}_{raw}} P_r R_{r,0} - \sum_{j\in\mathcal{J},T\in\mathcal{T}} F_{j,t} \tag{3.17}$$

$$s.t. \quad \begin{array}{ll} \text{Assignment constraints} & \text{Eq.(3.1)} \\ \text{Material balance} & \text{Eq.(3.2)} \\ \text{Capacity constraints} & \text{Eqs.(3.3)(3.4)} \\ \text{Timing constraints} & \text{Eqs.(3.5)(3.6)(3.7)} \\ \text{Material quality} & \text{Eqs.(3.8)(3.9)} \\ \text{Unit operation} & \text{Eq.(3.16)} \end{array}$$

**Tightening Constraints**

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

*Tightening timing constraints*

As studied in the context of conventional batch scheduling problems in linear form, timing constraints are a critical determinant of the tightness of the corresponding relaxed linear programs [120]. We keep the use of a good tightening constraint, which states that the summation of processing times of any unit over all event slots needs to be less or equal than the length of the scheduling horizon.

$$\sum_{t\in\mathcal{T}} Tp_{j,t} \leqslant H, \quad \forall j \in \mathcal{J}. \tag{3.18}$$

*Mass balance of a unit*

Although the production and consumption of materials are (generally nonlinear)

functions of state variables, the linear mass balance relationship still holds for each process unit. The rationale for preserving the mass balance equations is to assure the material conservation law for the relaxed problem, where the integrality requirement on binary variables is dismissed. In Eq. (3.19), balance equations are written with respect to individual units.

$$\sum_{r\in\mathcal{R}_j^p} R_{j,r,t}^p = \sum_{r\in\mathcal{R}_j^c} R_{j,r,t}^c, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}. \tag{3.19}$$

**Overall Formulation**

With all the elements at hand, the overall formulation is ready to be demonstrated. It presents as an MLDO problem, including Eqs. (3.17)-(3.19). There are two important factors that need attention when applying this formulation. First, as pointed out by Li and Floudas [123], a proper number of event points is conducive to balance between the model optimality and the computational complexity of an event-based scheduling formulation. Second, the scale of dynamic models employed in the formulation is of equal importance in determining its performance.

**Reduced Models of Recipe-based Operating States**

For batch units that do not require dynamic optimization in the SEN, the recipe-oriented strategy can still be accommodated by reducing dynamic models. As the process control inputs and processing times are fixed according to recipes, state profiles are only functions of initial conditions of differential variables. Therefore for a dynamic model of an operating state, its performance indices such as production, consumption and operating cost can be reduced to functions of the batch size and quality variables. These functions are often in simple algebraic form. Furthermore, quality measurements of materials become constants and constraints on quality measurements can be simplified or even discarded,

when all operation sequences are restricted by recipes. Under this circumstance, the batch size variables are the only ones left to quantify the performance of operating states, where linear equations are frequently encountered. This gives rise to a reduced problem without dynamic and quality variables. The reduced problem can be conceived as a typical recipe-based batch scheduling problem and recast as an MILP. It is also worth to note that any solution of the reduced problem completed with the constant values of the dynamic and quality variables is still a feasible solution of the corresponding full space problem. In fact, the best of these solutions is often a good initial guess to start with for solving the full problem.

### 3.2.4 Solution Strategy

The problem is solved with two steps. First, we reformulate MLDOs to MIDOs. There exist two ways to reformulate logical disjunctions, namely, the Big M method and the convex hull relaxation (CHR). The convex hull relaxation generally leads to a tighter relaxed problem at cost of a larger problem size, compared with the Big M method. Hence, the performance of these two approaches is problem dependent. In the integrated SEN formulation, however, dynamic models of operating states are generally nonlinear and nonconvex, such that applying the CHR is risky at cutting off the feasible region of an MLDO. The Big M method can avoid disaggregating continuous variables that mainly consist of the states and controls of dynamic models. Furthermore, we choose an enhanced version of the Big M method called the Multi-M reformulation. In this approach, nonidentical $M$s are allowed to be associated with different equations in a disjunct, and they are set to values as small as possible, while being sufficiently large to preserve equivalent feasibility. This technique provides a tighter reformulation than the conventional Big M approach without increasing the size of a problem [124]. Next, we convert the MIDO problem to the MINLP form using the simultaneous collocation strategy. In this

study, we accept good feasible solutions identified by local algorithms within reasonable solution times. Because the integrated formulation (3.17) has a relatively small number of binary variables but many highly nonlinear equations, we choose the NLP-based branch and bound approach (see Tab. 2.3). In addition, we develop an initialization phase to accelerate the optimization algorithm: it first starts with obtaining the optimal solution of the recipe-based case (often MILP), then a continuous nonlinear optimization of the integrated model is carried out via fixing production sequences (binary variables) to the recipe solutions. Finally, the integrated model is solved departing from the solution generated by the previous step that is consistent with dynamic models with controls fixed.

## 3.3 Case Studies

We test the integrated formulation in comparison with the recipe-based approach to show the benefits of incorporating dynamics into scheduling. In the recipe-based approach, batch recipes are given in advance which contain fixed values of control inputs and processing times of operating states. With these, we reduce the integrated models to their equivalent linear counterparts. The reduced models are recast and solved as MILPs (see Appendix A) to determine the corresponding optimal schedules for the recipe-based case. Nonetheless, the integrated models maintain full degrees of freedom in control, covering detailed dynamic and quality information. In this work, we create all the models in the GAMS [125] environment and apply CPLEX as the MILP solver, while SBB is used to solve MINLPs employing CONOPT as the nonlinear subsolver. These two examples are typical batch processes and the models are based on typical batch unit models such as in Bhatia and Biegler [46], as well as other literature references. For detailed information including scheduling parameters and unit dynamic models, please refer to Appendix A.

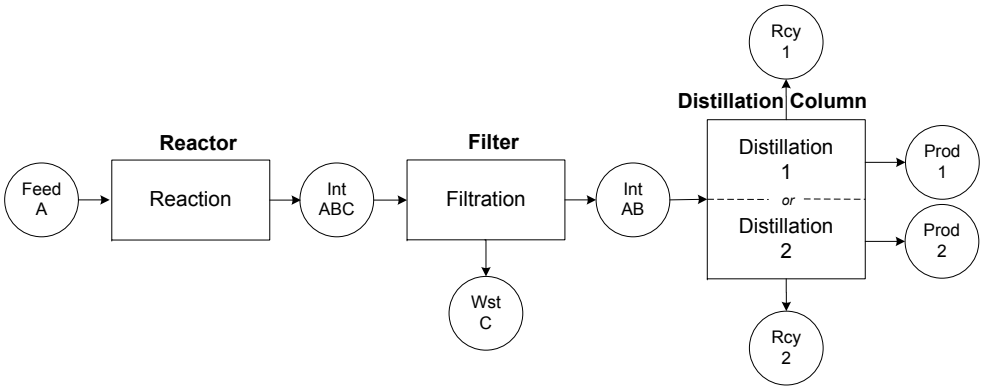



Figure 3.2: State equipment network for the flowshop plant

### 3.3.1 A Flowshop Plant

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

$$FeedA \xrightarrow{Reaction} IntABC \xrightarrow[\dashrightarrow WasteC]{Filtration} IntAB \xrightarrow[\dashrightarrow Recycle1]{Distillation1} Product1$$

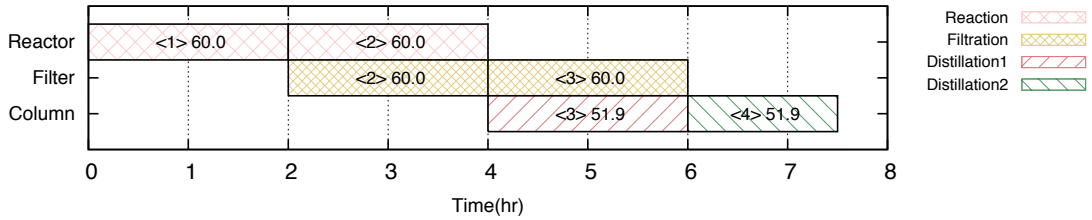
$$FeedA \xrightarrow{Reaction} IntABC \xrightarrow[\dashrightarrow WasteC]{Filtration} IntAB \xrightarrow[\dashrightarrow Recycle2]{Distillation2} Product2$$

To represent the flow shop using the SEN framework, there needs to be three units (*Reactor*, *Filter*, *Distillation Column*) with four operating states (*Reaction*, *Filtration*, *Distillation 1*, *Distillation 2*) and eight material states (*FeedA*, *IntABC*, *IntAB*, *WstC*, *Rcy1*, *Rcy2*, *Prod1*, *Prod2*) in the graph as shown in Fig. 3.2. For the two distillation states, *Distillation 1* lasts longer with a smaller constant reflux ratio. Thus, distillate *Product 2* is purer than *Product 1*. As a typical reaction-separation process, reaction and distillation states are recognized as the key stages where process dynamics can expect considerable benefits, so we embed dynamic models for all these stages and operate the filter without optimization of control. The time horizon of interest lasts for $7.5$ hours, and *Product 1* and *2* are produced once for each within the time window. The number of event slots are determined by running the recipe-based MILP model, where the total number of event slots increases from a small value until no improvement on the objective can been achieved by adding more

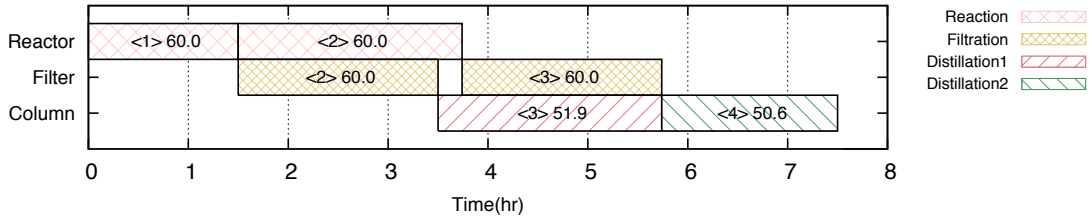
slots. In the integrated formulation, the same number of slots is used to align with the recipe-based model, though a larger number may improve the optimum. In order to approximate the dynamics accurately, eight finite elements are used inside each event slot, further with a two-point Radau collocation scheme.

We solve both the integrated and the recipe-based formulations and draw the optimal schedules as Gantt charts in Fig. 3.3 for comparison. Active operating states are depicted as patterned slots. In each slot, the number within brackets indicates the corresponding event number, and the following number is the batch size. In this example, the sequences of operations are the same for both formulations, but the timings are dissimilar. In both optimal schedules, the reaction and the distillation states are executed twice. However in the integrated case, the processing times show run-to-run variability in both dynamic operating states. In addition, the optimal control profiles of the two dynamic units are drawn in Figs. 3.4-3.5. The dotted lines sketch for the simple operating strategy specified by the recipe, while the solid lines represent the optimal dynamic curves obtained from the integrated model. It can be seen that an operating state may be executed differently in its individual appearances in the integrated model, such that the degrees of freedom in unit operations are fully explored by this method.

Some important data from the model statistics and solution are listed in Tab. 3.1. Because the same number of event slots is used, the two models have equal numbers of binary variables. However, the integrated model requires an additional group of continuous variables and constraints to account for the process dynamics that the recipe model does not take into account. These variables and constraints consequently increase the problem size and bring in nonlinearity and nonconvexity. Since the scale of this example is rather small, both models can be solved fully such that the gap between the original problem and its relaxation is closed. In the recipe case, the optimum is even found at the presolve stage by CPLEX. On the other hand, SBB explores 88 nodes and finds the best solution




(a) Optimal schedule of the recipe-based model




(b) Optimal schedule of the integrated model

Figure 3.3: Gantt charts of the flowshop plant

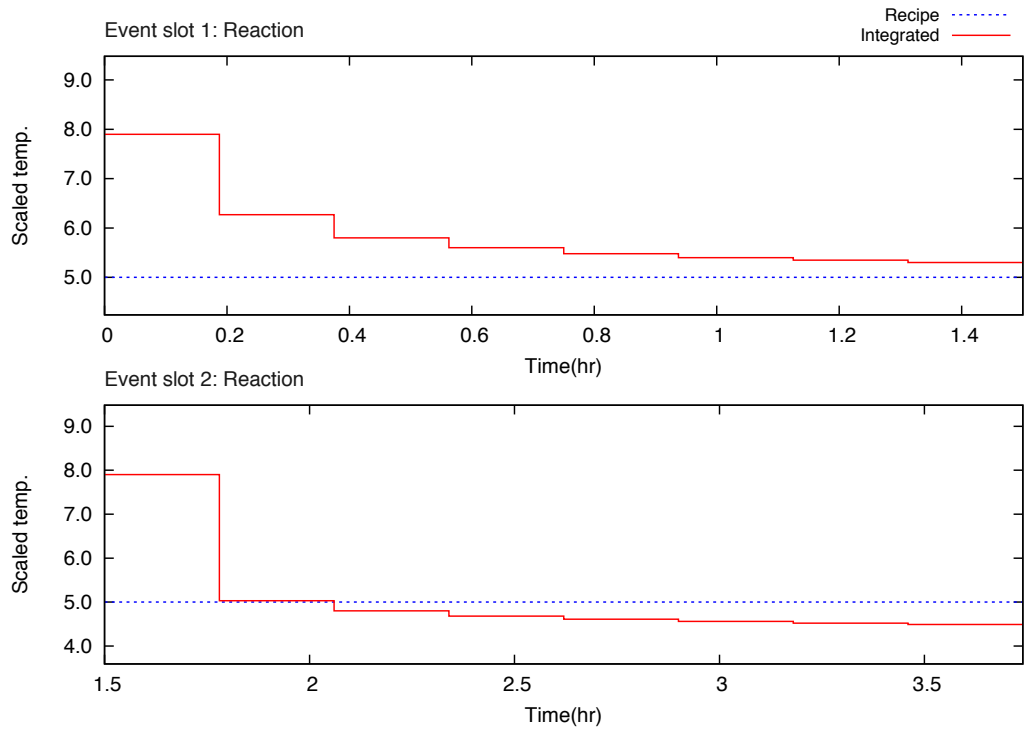



Figure 3.4: Optimal operating profiles for *Reactor* (flowshop)

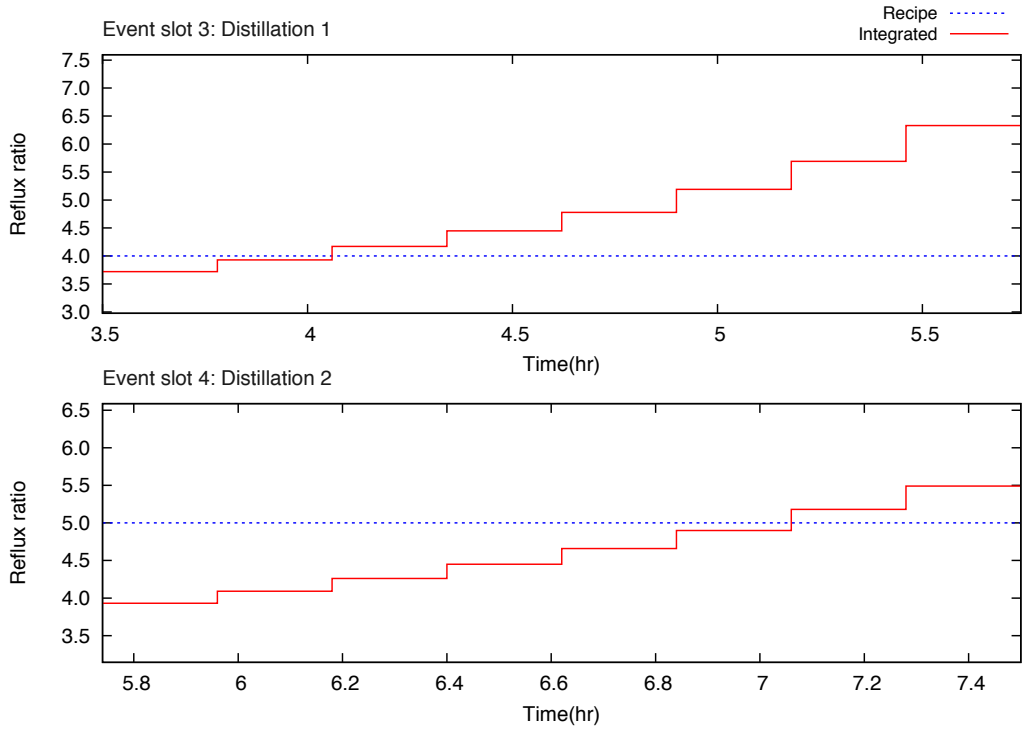



Figure 3.5: Optimal operating profiles for *Distillation Column* (flowshop)

at node $64$ for the integrated model. In terms of the objective value, the net profit gained by the integrated approach is $36.0\%$ higher. This improvement of profitability is very promising. As an outcome of the integrated decision-making approach, the reactor and the column are operated cooperatively. For instance, in the manufacturing sequence of *Product 2*, the reactor prepares a purer intermediate material compared with that of the recipe-based model by using a longer processing time ($2.24(hr)$ versus $2(hr)$[1]) and a dynamic temperature curve (Fig. 3.4). To be more specific, after removing waste component $C$ by the filter, the purity levels of intermediate product $IntAB$ are $90.7\%$ and $89.5\%$ (the concentration of component $B$) for the integrated and the recipe-based approaches, respectively. As a consequence, the distillation column is able to run with a $17.3\%$ longer processing time and a reflux profile that has a smaller average value than the recipe value (see Fig. 3.5), but the products still meet the quality requirement on purity, which is at

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

Table 3.1: Comparison of results for the flowshop example

| Model | Statistics | | | |
|---|---|---|---|---|
| | Type | Var.(Discrete) # | Nonlinear Var. # | Cons. # |
| Recipe-based | MILP | 153(16) | 0 | 499 |
| Integrated | MINLP | 1789(16) | 1240 | 4338 |
| Model | Solution | | | |
| | Profit($MU$) | CPU time ($s$) | Node(Best) # | Gap(%) |
| Recipe-based | 407 | 0.11 | 0(0) | 0.0 |
| Integrated | 554 | 46.92 | 88(64) | 0.0 |

least $99.7\%$ for both cases. In the integrated solution, more *Product 2* can be obtained from the distillation ($26.3(kg)$ versus $21.3(kg)$), though the batch size of the distillation operation is slightly less than that of the recipe-based schedule (see Fig. 3.3). To account for the manufacturing costs, the same amount of raw materials are consumed in both cases, while the operating costs are higher in the integrated case ($512(MU)$ versus $457(MU)$[2]). In this example however, the increment in product sales is more than compensates for the increased cost. There are probably other opportunities for trade-offs between product sales and manufacturing costs by adjusting operational level and scheduling level decisions. The integrated approach is able to take advantage of these opportunities, and therefore improves the profitability of the overall process considerably.

### 3.3.2 A Jobshop Plant

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

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

$$FeedA \xrightarrow{Reaction1} IntABC \xrightarrow[-\!\!\rightarrow WasteC]{Filtration1} IntAB \xrightarrow[-\!\!\rightarrow Recycle1]{Distillation1} Rprod1 \xrightarrow{Packaging1} Product1$$

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

$$FeedA \xrightarrow{Heating} HotA \xrightarrow{Reaction2} IntADE \xrightarrow[-\!\!\rightarrow WasteA]{Filtration2} IntDE \xrightarrow{Packaging3} Product3$$

Both reactors operate *Reaction 1* and *2*, and similarly, the packaging lines and the distillation column work for multiple purposes, but the heater and the filters have dedicated functions. Again, we embed dynamic models for the reaction and distillation states. The SEN representation of this problem is shown in Fig. 3.6. In this figure, parallel units that share the same set of operating states are depicted in a combined manner, since the possible connections are the same. In this example, schedules are designed to predict a time horizon of $10$ hours, within which six event points are defined. The optimal schedules of the recipe-based and the integrated approaches are compared in Fig. 3.7. The optimal operating profiles of the three dynamic units are depicted in Figs. 3.8-3.10, and important data are listed in Tab. 3.2. In the Gantt charts (Fig. 3.7), it can be seen that a larger number of operations are executed in the integrated case, e.g., one more slot for *Reaction 2* of *Reactor 1*. In the recipe-based schedule, *Packaging 2* occurs only once and *Packaging 3* takes place twice; but it is the opposite in the integrated schedule so that the production of *Product 2* is much higher in this case. As described earlier, the production sequence of *Product 2* requires an additional distillation operation than that of *Product 3*. As a result, *Product 2* is purer (*Product 2* : $99.3\%$ versus *Product 3* : $74.0\%$) and of a higher market price (*Product 2* : $50(MU/kg)$ versus *Product 3* : $20(MU/kg)$). The integrated schedule leads to a better production of *Product 2* that consequently contributes to the increase of profit (see Tab. 3.2), and one should notice that this schedule is not optimal nor even feasible for the recipe scheme. The result here shows when controls are allowed to vary, the design of batch schedules becomes more flexible. For this reason, one may obtain a product portfolio that differs from the recipe-based method, and the portfolio can be more profitable as well.

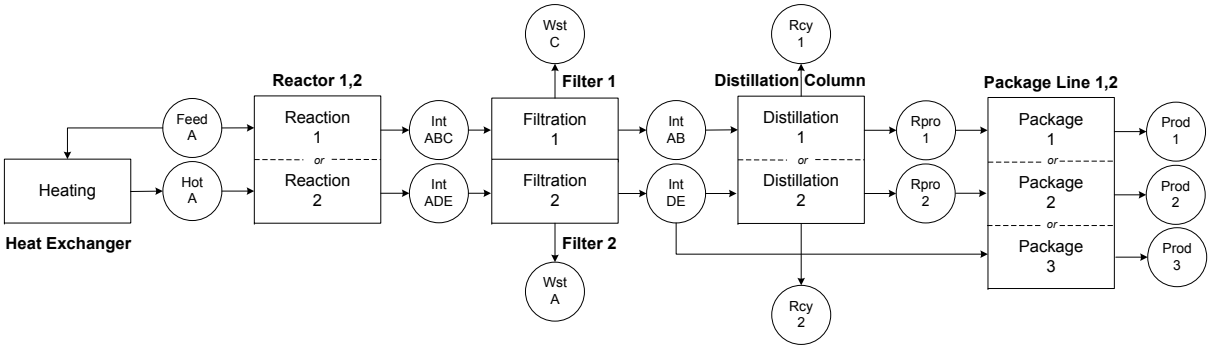
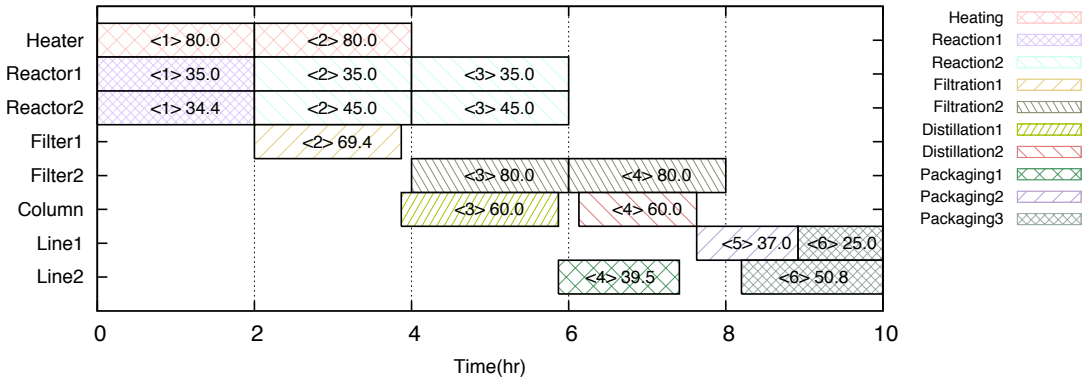
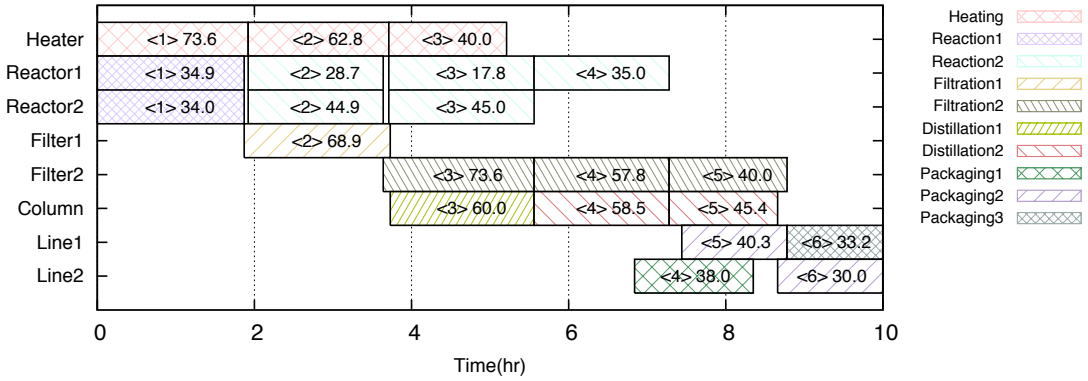



Figure 3.6: State equipment network for the jobshop plant




(a) Optimal schedule of the recipe-based model




(b) Optimal schedule of the integrated model

Figure 3.7: Gantt charts of the jobshop plant

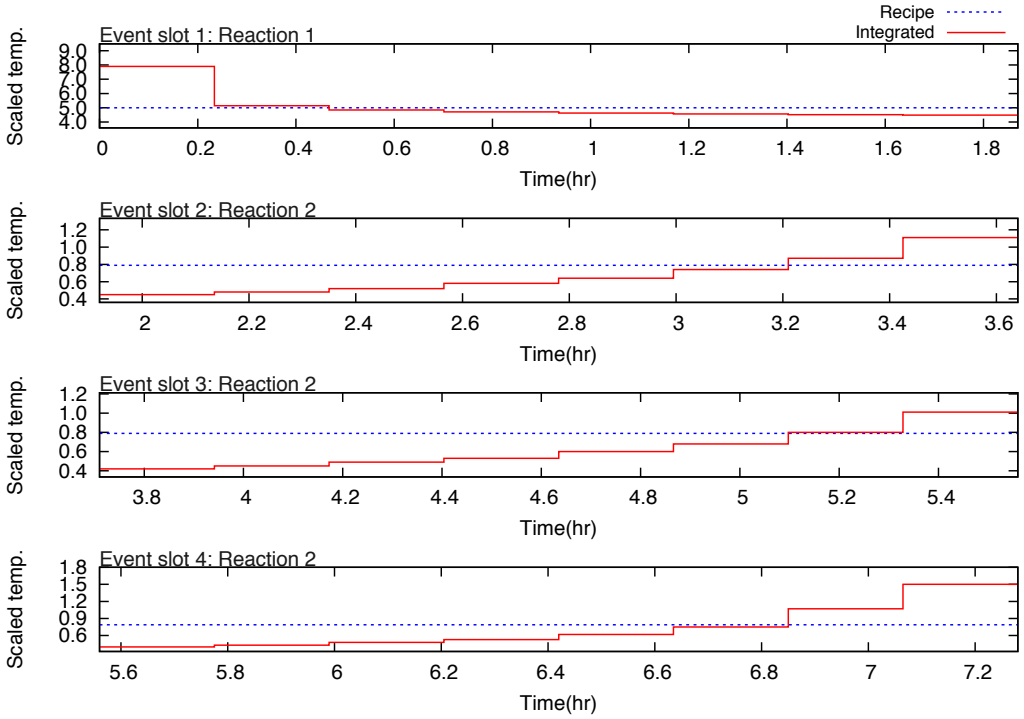



Figure 3.8: Optimal operating profiles for *Reactor 1* (jobshop)

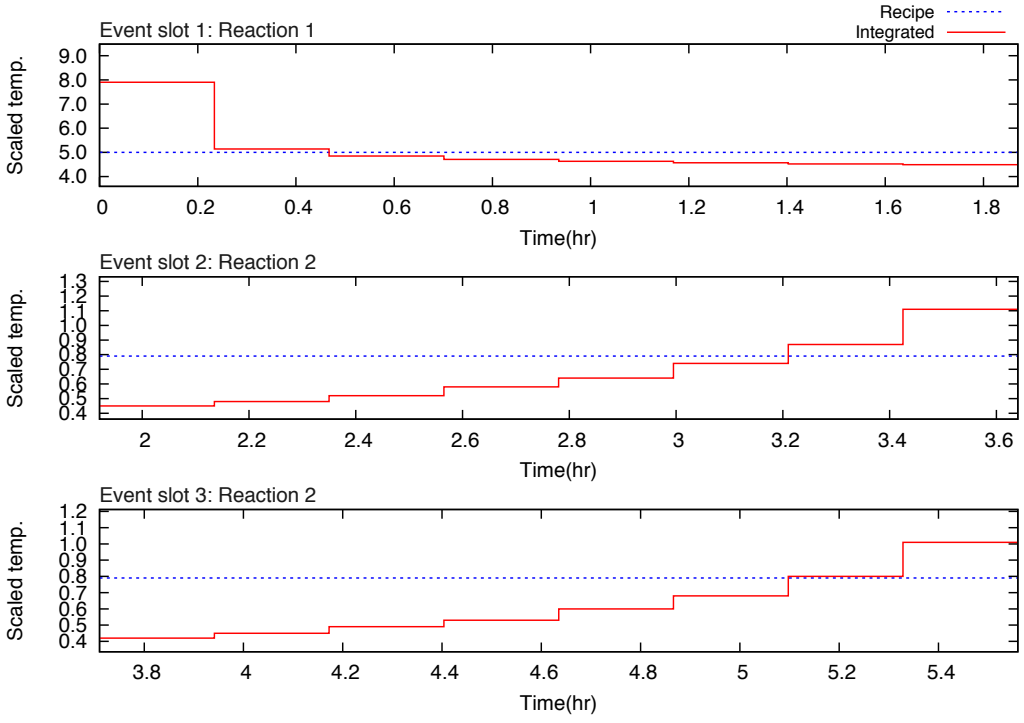



Figure 3.9: Optimal operating profiles for *Reactor 2* (jobshop)

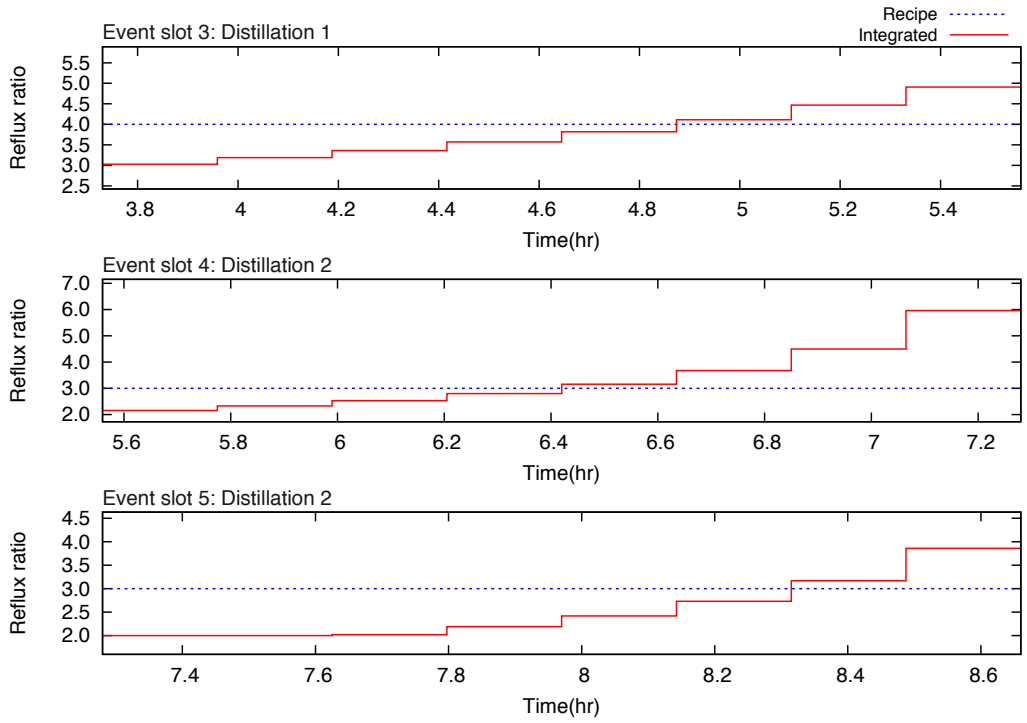



Figure 3.10: Optimal operating profiles for *Distillation Column* (jobshop)

Table 3.2: Comparison of results for the jobshop example

| Model | Statistics | | | |
|---|---|---|---|---|
| | Type | Var.(Discrete) # | Nonlinear Var. # | Cons. # |
| Recipe-based | MILP | 676(90) | 0 | 1079 |
| Integrated | MINLP | 4978(90) | 2292 | 12507 |
| **Model** | **Solution** | | | |
| | Profit($MU$) | CPU time ($s$) | Node(Best) # | Gap(%) |
| Recipe-based | 1374 | 0.366 | 288(199) | 0.0 |
| Integrated | 1935 | 9564 | 5000(1602) | 67.9 |

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

## 3.4 Concluding Remarks

The integrated formulation for scheduling and dynamic optimization has been developed within the state equipment network framework that serves for multiproduct and multipurpose batch plants. The basic elements of the SEN we use in this work are identified as *material state*, *equipment* and *operating state*. With this definition, we focus on the allocation of operating states over the scheduling horizon and process units to design batch schedules and provide optimal control strategies of active operating states. The proposed formulation is tested on flowshop and jobshop case studies and shows advantages over

the recipe-based approach, because dynamic optimization of unit operations leads to better overall performance of batch units.

This study shows strong evidence of economic benefits that the integration strategy is able to deliver to chemical plant operations. This motivates us to investigate industrial problems that represent the current practice in the chemical industry. The task is challenging due to additional complexity coming from more realistic scheduling constraints and sophisticated dynamic modes. However, we expect advanced optimization strategies are able to make improvements to current practice in the industry.

## 3.5 SEN Model Notation

| *Indices* | |
|---|---|
| $j$ | $j \in \mathcal{J}$ units |
| $r$ | $r \in \mathcal{R}$ materials |
| $s$ | $s \in \mathcal{S}$ operating states |
| $t$ | $t \in \mathcal{T}$ event slots/points (staring points of event slots) |
| $c$ | $c \in \mathcal{C}$ components |
| $k$ | $k \in \mathcal{K}$ finite elements |
| $q$ | $q \in \mathcal{Q}$ collocation points |
| *Sets* | |
| $\mathcal{J}$ | units |
| $\mathcal{R}$ | materials |
| $\mathcal{R}_{prod}$ | final products |
| $\mathcal{R}_{raw}$ | raw materials |
| $\mathcal{S}$ | operating states |
| $\mathcal{T}$ | event slots |
| $\mathcal{J}_s$ | units that have operating state $s$ |
| $\mathcal{J}_r^p$ | units that produce resource $r$ |
| $\mathcal{J}_r^c$ | units that consume resource $r$ |
| $\mathcal{R}_s^p$ | materials produced in operating state $s$ |
| $\mathcal{R}_s^c$ | materials consumed in operating state $s$ |
| $\mathcal{R}_j^p$ | materials produced in unit $j$ |
| $\mathcal{R}_j^c$ | materials consumed in unit $j$ |
| $\mathcal{S}_j$ | operating states that appear in unit $j$ |
| $\mathcal{S}_r^p$ | operating states that produce material $r$ |
| $\mathcal{S}_r^c$ | operating states that consume material $r$ |
| $\mathcal{C}_r$ | components of material $r$ |
| *Parameters* | |
| $T$ | cardinality of set $\mathcal{T}$ |
| $P_r$ | price of material $r$ |
| $H$ | scheduling horizon |
| $B_j^{max}, B_j^{min}$ | upper and lower bound of batch size for unit $j$ |
| $E_r^{max}, E_r^{min}$ | upper and lower bound of excess material $r$ |
| $\overline{E}_r$ | initial amount of material $r$ |
| $\eta_{r,c,1}$ | initial concentration distribution of $r$ |
| $\overline{\eta}_{r,c}$ | quality requirement for excess material $r$ |
| $\overline{\phi}_{j,r,c}$ | quality requirement for material $r$ produced by unit $j$ |

| *Variables* | |
|---|---|
| $w_{j,s,t}$ | binary variable indicate state of unit $j$ within event slot $t$ |
| $F_{j,t}$ | operating cost of unit $j$ within event slot $t$ |
| $E_{r,t}$ | amount of excess material at event point $t$ |
| $E_{r,0}$ | amount of material $r$ at the beginning of time horizon |
| $R^p_{j,r,t}$ | production of $r$ by unit $j$ at the end of slot $t$ |
| $R^c_{j,r,t}$ | consumption of $r$ by unit $j$ at event point $t$ |
| $b_{j,t}$ | batch size of unit $j$ in state $s$ within event slot $t$ |
| $T_{j,t}$ | beginning time of event slot $t$ |
| $Tp_{j,t}$ | processing times of unit $j$ within event slot $t$ |
| $\tau$ | normalized time |
| $\eta_{r,c,t}$ | fraction of component $c$ in $r$ at event point $t$ |
| $\phi_{j,r,c,t}$ | fraction of component $c$ in $r$ produced by $j$ at the end of $t$ |
| *Superscripts* | |
| $min$ | minimum |
| $max$ | maximum |
| $p$ | production |
| $c$ | consumption |

# Part II

# Industrial Applications on Polymerization Processes

**A Short Disclaimer**

This study aims at developing a systematic optimization framework for real-world processes, following the concept of integrated scheduling and dynamic optimization. Therefore, it includes basic components such as polymerization reactor modeling and scheduling algorithm development, as well as the integrated optimization method. The work in collaboration with The Dow Chemical Company. The example processes used in the following chapters are modified from one of Dow Chemical's polymer processes for optimization method development, and they do not represent actual operation practice.

# Chapter 4

# Reactor Model Development and Homopolymerization Recipe Optimization

In this chapter, we develop rigorous first-principles models for semi-batch polyether polyol reactors, and also perform dynamic optimization to optimize the reactor operation. The scope of the problem is limited at the unit operation level: a single reactor with individual polymerization runs. The development of dynamic reactor models pave the way for the future development of integrated optimization strategies. The reactor model is based on first-principles including the mass and population balances, reaction kinetics and vapor-liquid equilibria. Next, the obtained differential algebraic model is reformulated by applying a nullspace projection method that results in an equivalent dynamic system with better computational performance. The reactor model is validated against plant data by adjusting model parameters. In the recipe optimization dynamic optimization problem, the polymerization time is minimized, given a target product molecular weight as well as other requirements on product quality and process safety. The dynamic

optimization problem is solved by using the simultaneous collocation strategy and the interior point method. A case study example for polypropylene glycol polymerization is discussed. The result shows a good match between the model prediction and real plant data for process simulation, and the optimization approach is able to significantly reduce the batch time by $47\%$.

## 4.1 Background Information

Polyether polyols serve as important raw materials in the urethane industry, which represents roughly five percent of the worldwide polymer consumption [126]. Around ninety percent of all flexible foams produced today are made from polyether type polyols [127]. Other important applications of polyether polyols include polyglycols and surfactants. Similar to other commodity polymers, product quality is measured in many aspects and quantified in indices such as molecular weight (MW) and polydispersity index (PDI). Commercial alkoxylation processes are usually conducted through the reaction of alkylene oxides (e.g. ethylene oxide (EO) and/or propylene oxide (PO)) with starters (a.k.a. initiators) containing active hydrogen atoms (e.g. alcohols, amines, or even water). In practice, the polymerization process is catalyzed by a basic catalyst such as potassium hydroxide (KOH) at temperatures above $100\ ^{\circ}\mathrm{C}$. A typical example is the anionic polymerization reaction of PO, the modeling of which has been extensively studied from both academic and industrial [127] perspectives. Guibert et al. [128] and Di Serio et al. [129] studied the kinetics of propoxylation processes catalyzed by KOH. In the latter, a kinetic model addressing the initiation, propagation and cation exchange reactions was developed as well as vapor-liquid equilibrium (VLE) relations based on modified Raoult's law and Wilson equations. However, both works have not considered the effect of the proton transfer reaction, which gives rise to a small amount of unsaturated (unsat) monofunctional polymer chains with allyl and propenyl end groups. This reaction leads to im-

purity in the product polyols that is detrimental for further synthesis of polyurethanes. The transfer reaction was investigated in 1960 by Simons and Verbanc [130] and more recently by Heatley et al. [131]. Their research shows that the transfer reaction is typically two orders of magnitude slower than the propagation reaction and is suppressed by hydrogen bonding of hydroxyl groups to active ion pairs. From the industrial point of view, Wegener et al. [132] discussed the use of alternative catalyst systems to reduce the concentration of unsaturated chains and proposed a formula for estimating the actual functionality of polyether polyols taking into account the catalyst type. In addition, Di Serio et al. [133] compared different reactor types commonly adopted in industry for ethoxylation and/or propoxylation and the key factors examined included productivity, energy efficiency and safety. Although widely practiced in experimental research and industrial manufacturing, a comprehensive dynamic reactor model of the propoxylation process, especially with accurate quantitative description of the unsat chain population, is desired for process analysis and technology improvement. We first develop a comprehensive first-principles process model in terms of conservation laws, reaction kinetics and phase equilibria, which results in a system of DAEs. Using the developed model, mathematical programming techniques can be conveniently carried over to optimize the process performance. However, detailed modeling of polymerization processes generally leads to large-scale models, for which the computational issue should be carefully addressed, particularly for the sake of optimization. Owing to the advance of dynamic optimization techniques, many successful applications haven been reported to improve various polymer product categories, e.g., low-density polyethylene [134], high-impact polystyrene [90], gas-phase polyolefin [135, 136], polyurethane [137], and seeded suspension styrene polymerization [138]. The underlying principle of these research studies is to use mechanistic models based on first principles to predict the dynamic behavior of the process and optimize the process performance by adjusting process control decisions guided by solving associated dynamic optimization problems.

## 4.2 Reactor Model Development

The propoxylation process is a semi-batch process that can be carried out in a conventional stirred-tank reactor equipped with heat exchangers for heating and cooling. The starter is formed by mixing the alcohol and catalyst in an appropriate ratio; and in certain applications, water is also added which causes hydrolysis of alkylene oxides to form additional alcohols. After the starter is generated, the monomer is fed into the reactor continuously to grow polymers. External heat is required in the start-up stage and soon after the polymerization reactions have been kicked off, a significant amount of heat is released from the reactions and needs to be removed from the tank.

### 4.2.1 Reaction Mechanism

During the anionic polymerization process, each polymer chain undertakes the initiation, propagation, cation exchange and proton transfer reactions. A chain is started when an alkaline anion first reacts with PO, and then the resulting oxy-propylene anion can undertake propagation steps by successively adding monomers through the propagation reaction. The reactivities of chains are affected by the functional end groups; that is, chains ended by hydroxyl groups become dormant while chains with potassium ion can preserve high activity. During polymerization, exchanges of the end groups between species are observed and it is well known that these reversible reactions are very fast, such that the equilibrium does not influence the polymerization, beyond ensuring that all hydroxyl groups in the system serve as equivalent sites of propagation [131]. Lastly, unsaturated byproducts are formed due to the tendency for rearrangement of PO to allyl alcohol. The process can be classified as a living polymerization system, since the transfer reaction does not terminate chain growth and each chain retains the ability to undertake infinite propagation owing to the presence of the exchange reactions. However, it is worth noting

that the transfer reaction creates new chains that live for different but shorter periods than the initial ones.

The following notation is used in the remainder of the article. Let $\mathrm{M}$ denote the monomer PO. Let $\mathrm{P_n}$ denote the chain $\mathrm{CH_3(PO)_n}$, which comprises one of the branches of a polymeric alcohol and $n$ indicates the number of the repeating unit. Meanwhile, $\mathrm{U_n}$ represents the unsat chains with double-bond end groups $\mathrm{CH_2 = CHCH_2(PO)_n}$. In addition, $\mathrm{W}$ is introduced to account for the presence of water in the initial charge. Furthermore, depending on the different functional end groups, we define:

- $\mathrm{G_n}$ to denote the growing product chains of length $n$ ($\mathrm{P_nO^-K^+}$);
- $\mathrm{D_n}$ to denote the dormant product chains of length $n$ ($\mathrm{P_nOH}$);
- $\mathrm{Q_n}$ to denote the growing unsat chains of length $n$ ($\mathrm{U_nO^-K^+}$);
- $\mathrm{R_n}$ to denote the dormant unsat chains of length $n$ ($\mathrm{U_nOH}$).

The reaction schemes can be therefore summarized in Tab. 4.1. In this work, the kinetic parameters for the product and unsat chains are assumed to be identical for all reactions. Moreover, the asymmetric characteristic of the PO molecule may produce both primary and secondary alcohols in the initiation step, but the latter are found to be dominant [139]. Next in the propagation reaction, the ring-opening insertion of PO can be conducted by either head-to-head, head-to-tail, or tail-to-tail additions (here *head* refers to $\mathrm{CH(CH_3)}$ group and *tail* refers to $\mathrm{CH_2}$ group). It is also shown by Heatley et al. [140] that the head-to-tail placement prevails and its proportion is normally above ninety percent. In this study, the minor reactions stated above are ignored for simplicity. For the transfer reaction, note that the isomerization of allyl end groups to propenyl end groups is not discussed in this work. Finally the acid-base proton exchange reactions take place within and across the major and minor populations, and these reactions are reversible and approach equilibrium states. They are the fastest reactions in anionic alkoxylation, and their equilibrium constants are around unity since the acidity of the participating species is similar [141]. Furthermore, the exchange reaction between $\mathrm{G}$ and $\mathrm{D}$ is expressed with a single reaction

| | |
|---|---|
| Hydrolysis: | |
| | $\mathrm{W} + \mathrm{M} \xrightarrow{k_\mathrm{h}} 2\mathrm{D}_0$ |
| Initiation: | |
| | $\mathrm{G}_0 + \mathrm{M} \xrightarrow{k_\mathrm{i}} \mathrm{G}_1$ |
| | $\mathrm{Q}_0 + \mathrm{M} \xrightarrow{k_\mathrm{i}} \mathrm{Q}_1$ |
| Propagation: | |
| | $\mathrm{G}_\mathrm{n} + \mathrm{M} \xrightarrow{k_\mathrm{p}} \mathrm{G}_{\mathrm{n}+1} \quad (\mathrm{n} \geqslant 1)$ |
| | $\mathrm{Q}_\mathrm{n} + \mathrm{M} \xrightarrow{k_\mathrm{p}} \mathrm{Q}_{\mathrm{n}+1} \quad (\mathrm{n} \geqslant 1)$ |
| Transfer: | |
| | $\mathrm{G}_\mathrm{n} + \mathrm{M} \xrightarrow{k_\mathrm{t}} \mathrm{D}_\mathrm{n} + \mathrm{Q}_0 \quad (\mathrm{n} \geqslant 0)$ |
| | $\mathrm{Q}_\mathrm{n} + \mathrm{M} \xrightarrow{k_\mathrm{t}} \mathrm{R}_\mathrm{n} + \mathrm{Q}_0 \quad (\mathrm{n} \geqslant 0)$ |
| Exchange: | |
| | $\mathrm{G}_\mathrm{n} + \mathrm{D}_\mathrm{m} \xrightarrow{k_\mathrm{e}} \mathrm{D}_\mathrm{n} + \mathrm{G}_\mathrm{m} \quad (\mathrm{n}, \mathrm{m} \geqslant 0)$ |
| | $\mathrm{Q}_\mathrm{n} + \mathrm{R}_\mathrm{m} \xrightarrow{k_\mathrm{e}} \mathrm{R}_\mathrm{n} + \mathrm{Q}_\mathrm{m} \quad (\mathrm{n}, \mathrm{m} \geqslant 0)$ |
| | $\mathrm{G}_\mathrm{n} + \mathrm{R}_\mathrm{m} \underset{k_e}{\overset{k_e}{\rightleftharpoons}} \mathrm{D}_\mathrm{n} + \mathrm{Q}_\mathrm{m} \quad (\mathrm{n}, \mathrm{m} \geqslant 0)$ |

Table 4.1: Reactions in anionic PO polymerization

rate, because the reactants and products are symmetric, which also applies for $\mathrm{Q}$ and $\mathrm{R}$. However, the cross-population exchange reaction needs two rates to describe it.

### 4.2.2 First-principles Model

The first-principles model consists of population balance equations of polymer chains and monomers, overall mass balances and liquid density correlations, as well as VLE equations. Reactor temperature and monomer feed rate are time-dependent operating decisions, rendering degrees of freedom for process recipe design. According to the reaction schemes described earlier, the population balance equations for individual species

can be established as follows:

$$\frac{\mathrm{d}(V[\mathrm{W}])}{\mathrm{d}t} = -Vk_h[\mathrm{W}][\mathrm{M}], \tag{4.1a}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{G_0}])}{\mathrm{d}t} =& -Vk_i[\mathrm{G_0}][\mathrm{M}] - Vk_t[\mathrm{G_0}][\mathrm{M}] - Vk_e[\mathrm{G_0}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) \\ &+ Vk_e[\mathrm{D_0}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]),\end{aligned} \tag{4.1b}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{G_1}])}{\mathrm{d}t} =& V(k_i[\mathrm{G_0}] - k_p[\mathrm{G_1}])[\mathrm{M}] - Vk_t[\mathrm{G_1}][\mathrm{M}] - Vk_e[\mathrm{G_1}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) \\ &+ Vk_e[\mathrm{D_1}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]),\end{aligned} \tag{4.1c}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{G_n}])}{\mathrm{d}t} =& Vk_p([\mathrm{G_{n-1}}] - [\mathrm{G_n}])[\mathrm{M}] - Vk_t[\mathrm{G_n}][\mathrm{M}] - Vk_e[\mathrm{G_n}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) \\ &+ Vk_e[\mathrm{D_n}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]), \qquad n = 2, \dots N-1,\end{aligned} \tag{4.1d}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{G_N}])}{\mathrm{d}t} =& Vk_p([\mathrm{G_{N-1}}])[\mathrm{M}] - Vk_t[\mathrm{G_N}][\mathrm{M}] - Vk_e[\mathrm{G_N}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) \\ &+ Vk_e[\mathrm{D_N}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]),\end{aligned} \tag{4.1e}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{D_0}])}{\mathrm{d}t} =& V2k_h[\mathrm{W}][\mathrm{M}] + Vk_t[\mathrm{G_0}][\mathrm{M}] + Vk_e[\mathrm{G_0}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) \\ &- Vk_e[\mathrm{D_0}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]),\end{aligned} \tag{4.1f}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{D_n}])}{\mathrm{d}t} =& Vk_t[\mathrm{G_n}][\mathrm{M}] + Vk_e[\mathrm{G_n}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) \\ &- Vk_e[\mathrm{D_n}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]), \qquad n = 1, \dots N,\end{aligned} \tag{4.1g}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{Q_0}])}{\mathrm{d}t} =& -Vk_i[\mathrm{Q_0}][\mathrm{M}] + Vk_t\sum_{n=0}^{N}([\mathrm{G_n}] + [\mathrm{Q_n}])[\mathrm{M}] - Vk_t[\mathrm{Q_0}][\mathrm{M}] \\ &- Vk_e[\mathrm{Q_0}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) + Vk_e[\mathrm{R_0}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]),\end{aligned} \tag{4.1h}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{Q_1}])}{\mathrm{d}t} =& V(k_i[\mathrm{Q_0}] - k_p[\mathrm{Q_1}])[\mathrm{M}] - Vk_t[\mathrm{Q_1}][\mathrm{M}] - Vk_e[\mathrm{Q_1}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) \\ &+ Vk_b[\mathrm{R_1}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]),\end{aligned} \tag{4.1i}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{Q_n}])}{\mathrm{d}t} =& Vk_p([\mathrm{Q_{n-1}}] - [\mathrm{Q_n}])[\mathrm{M}] - Vk_t[\mathrm{Q_n}][\mathrm{M}] - Vk_e[\mathrm{Q_n}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) \\ &+ Vk_e[\mathrm{R_n}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]), \qquad n = 2, \ldots, N-1,\end{aligned} \tag{4.1j}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{Q_N}])}{\mathrm{d}t} =& Vk_p([\mathrm{Q_{N-1}}])[\mathrm{M}] - Vk_t[\mathrm{Q_N}][\mathrm{M}] - Vk_e[\mathrm{Q_N}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) \\ &+ Vk_e[\mathrm{R_N}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]),\end{aligned} \tag{4.1k}$$

$$\begin{aligned}\frac{\mathrm{d}(V[\mathrm{R_n}])}{\mathrm{d}t} =& Vk_t[\mathrm{Q_n}][\mathrm{M}] + Vk_e[\mathrm{Q_n}]\sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) \\ &- Vk_e[\mathrm{R_n}]\sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]), \qquad n = 0, 1, \ldots N.\end{aligned} \tag{4.1l}$$

Here, $V$ is the volume of the liquid in the reactor and $[\cdot]$ denotes the liquid phase concentration. Moreover, to make computation tractable, a sufficiently large number $N$ is chosen to denote the length of the longest chains that are recorded in the model and chains beyond $N$ are ignored. In addition, chains with $N$ repeating units are assumed not to undertake propagation reactions. Commercial polyether type polyols typically have chain lengths less than $100$ [142] so that the resultant model sizes remain manageable even when the detailed population spectrum is calculated. For the monomer balance, the external feed is entering the reactor at rate $F$, and the monomers are consumed by all four reactions (hydrolysis, initiation, propagation and transfer):

$$\frac{\mathrm{d}(V[\mathrm{M}])}{\mathrm{d}t} = F - V(k_h[\mathrm{W}] + k_i([\mathrm{G_0}] + [\mathrm{Q_0}]) + k_p\sum_{n=1}^{N-1}([\mathrm{G_n}] + [\mathrm{Q_n}]) + k_t\sum_{n=0}^{N}([\mathrm{G_n}] + [\mathrm{Q_n}]))[\mathrm{M}] \tag{4.2}$$

For the total mass balance, as the monomer enters the system, we have:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = FMW_{\mathrm{PO}}, \tag{4.3}$$

where $MW_{\mathrm{PO}}$ denotes the molecular weight of PO and $m$ is the total mass of the polymerization system. The liquid density is solely dependent on the reactor temperature since the effects of molecular weight on density are found to be minor. Therefore, the liquid volume is calculated by [143]:

$$V = m(10^{-6} + 7.576 \times 10^{-10}(T - 298.15)). \tag{4.4}$$

Although the polymerization reactions occur only in the liquid phase, a faithful VLE model is still important for the purpose of process monitoring in manufacturing practice. Reactor pressure measures are often convenient to obtain, and can be used to imply the unreacted PO concentration in the liquid, which is difficult and also risky to measure directly. For a dummy volatile component $i$, the basic equation of the vapor-liquid phase equilibrium is written as

$$P_i = a_i P_i^{sat}, \tag{4.5}$$

where, $P_i$, $a_i$ and $P_i^{sat}$ are the partial pressure, liquid phase activity and saturated vapor pressure of component $i$, respectively. The total pressure of the reactor $P$ can be obtained by

$$P = \sum_i P_i. \tag{4.6}$$

Typically, PO is considered to be volatile, and other possible volatile components include water and starters. However, propylene oxide polyols are liquids in the molecular weight range of 200-6000 [143]. It follows the assumption that no polymers exist in the vapor phase. In addition, Eq. (4.6) may be adjusted in the presence of nitrogen in the reactor by also including the partial pressure over this non-volatile component. The vapor pressure

can be calculated by using the Antoine equation:

$$\log_{10} P_i^{sat} = A_i - \frac{B_i}{T + C_i}. \tag{4.7}$$

For the polymer-solvent equilibrium, the Flory-Huggins theory [144] provides a rational method to develop an expression for the activity of a solvent in a polymer. In this study, the liquid mixture contains multi-component solvents that complicates the calculation. A simple yet effective approximation is to treat the system as a pseudo binary mixture, where the solvent is PO (denoted in index $s$) and the other components are assumed to be the polymer (denoted in index $p$). The simplification is reasonable since by the time polyols are present in sufficient amounts to dominate VLE, both initiator and water have been almost fully reacted. Hence, the activity of PO is equal to:

$$\ln a_s = \ln \phi_s + (1 - \frac{1}{l})\phi_p + \chi \phi_p^2. \tag{4.8}$$

In Eq. (4.8), rather than using mole fractions, the fractions of lattice sites occupied by the solvent molecule $\phi_s$ and polymer $\phi_p$ are applied. The interaction parameter $\chi$ is non-dimensional and accounts for the energy of interdispersing polymer and solvent molecules. While polyols are small polymers compared to many other commercial ones, the effect of the number average chain length $l$ still needs to be taken into account when calculating the lattice fraction, shown as below:

$$\phi_s = \frac{n_s}{n_s + n_p l}; \tag{4.9a}$$

$$\phi_p = \frac{n_p l}{n_s + n_p l}. \tag{4.9b}$$

Here, $n_s$ and $n_p$ represent the numbers of molecules of the solvent (PO) and the polymer, respectively, calculated as

$$n_s = V[\mathrm{M}]; \tag{4.10a}$$

$$n_p = V([\mathrm{W}] + \sum_{n=0}^{N}([\mathrm{G_n}] + [\mathrm{D_n}] + [\mathrm{Q_n}] + [\mathrm{R_n}])). \tag{4.10b}$$

On the other hand, the activities of other volatile components such as water can be treated as constants for simplicity. To this end, the first-principles reactor model for describing the propoxylation process is complete. This model comprises differential and algebraic equations, and the involved differential and algebraic state variables can reveal detailed information of the system. The population distribution for all species is recoded in a chain length basis over the operation time horizon.

### 4.2.3 Reformulation of the Exchange Reactions

Synergistic fast and slow dynamic modes are often encountered in modeling chemical dynamic systems and cause difficulties in their numerical solution. Therefore, a reformulation procedure is required but it is often non-trivial to obtain by intuition, giving correct asymptotic characteristics. The reformulation and model reduction of such systems have been investigated by Daoutidis and coworkers, particularly in the context of solvent recycles [145]. The underlying idea is to separate the fast and the slow components in a DAE system by describing the fast ones with algebraic equations capturing their quasi-steady states. As a result, the reformulated system becomes less stiff but with the same asymptotic behavior. As noted earlier, among the polymerization reactions, the rates of the exchange reactions are significantly higher than those of the other reactions. Modeling the polymerization system consequently leads to a two-time-scale model and incurs the stiffness issue of the resulting DAE model derived in Eqs. (4.1a)-(4.1l). To address this

challenge, a nullspace projection method is discussed in the sequel, following a similar idea as the aforementioned one.

### Nullspace Projection

We develop a systematic reformulation procedure of reaction equation systems that is based on a nullspace projection method. Here, we consider a general reaction system given by:

$$\dot{x} = Ar(x) + g(t), \tag{4.11}$$

where $x \in \mathbb{R}^{n_x}$ is the vector of component concentrations/populations, $r(x) \in \mathbb{R}^{n_r}$ is the vector of reaction rates, $A \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_r}$ is the coefficient matrix, and $g(t) \in \mathbb{R}^{n_x}$ represents the external input such that $g(t) \not\equiv 0$ for fed-batch reactions and $g(t) \equiv 0$ for batch reactions. For the reactions, we partition $r(x)$ so that reactions that reach equilibrium are separated from those that do not, and the coefficients in matrix $A$ are partitioned accordingly, written as:

$$\dot{x} = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} r_1(x) \\ \sigma r_2(x) \end{bmatrix} + g(t). \tag{4.12}$$

In the equation, $\sigma r_2(x)$ represents the rates of equilibrium reactions, where $\sigma$ is a large positive number that can approach infinity. Next, a nullspace matrix $\mathcal{Z}$ is introduced such that $\mathcal{Z}^{\mathrm{T}} A_2 = 0$, and a corresponding matrix $\mathcal{Y}$ spanning the range space of $A_2$ is also defined, ensuring $\begin{bmatrix} \mathcal{Y} & \mathcal{Z} \end{bmatrix} \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ is non-singular. Multiplying the transposed matrix $\begin{bmatrix} \mathcal{Y} & \mathcal{Z} \end{bmatrix}^{\mathrm{T}}$ to the two sides of Eq. (4.12) gives:

$$\mathcal{Y}^{\mathrm{T}} \dot{x} = \mathcal{Y}^{\mathrm{T}} A_1 r_1(x) + \sigma \mathcal{Y}^{\mathrm{T}} A_2 r_2(x) + \mathcal{Y}^{\mathrm{T}} g(t), \tag{4.13a}$$

$$\mathcal{Z}^{\mathrm{T}} \dot{x} = \mathcal{Z}^{\mathrm{T}} A_1 r_1(x) + \mathcal{Z}^{\mathrm{T}} g(t). \tag{4.13b}$$

Eq. (4.13b) does not include equilibrium rates, and it is kept as a part of the reformulated system. On the other hand, the matrices in Eq. (4.13a) can be further rearranged and partitioned. On the right-hand-side, $\sigma \mathcal{Y}^{\mathrm{T}} A_2 r_2(x)$ corresponds to the effect of equilibrium rates and it can be separated into a zero and a non-zero part, and $\mathcal{Y}$ can be partitioned accordingly. Let $f(x)$ denote the non-zero elements in $\mathcal{Y}^{\mathrm{T}} A_2 r_2(x)$, and partitioning these elements leads to

$$\begin{bmatrix} \mathcal{Y}_a^{\mathrm{T}} \\ \mathcal{Y}_b^{\mathrm{T}} \end{bmatrix} \dot{x} = \begin{bmatrix} \mathcal{Y}_a^{\mathrm{T}} \\ \mathcal{Y}_b^{\mathrm{T}} \end{bmatrix} A_1 r_1(x) + \begin{bmatrix} 0 \\ \sigma f(x) \end{bmatrix} + \begin{bmatrix} \mathcal{Y}_a^{\mathrm{T}} \\ \mathcal{Y}_b^{\mathrm{T}} \end{bmatrix} g(t). \tag{4.14}$$

When $\sigma \to \infty$, the rows corresponding to zeros in the second term on the right-hand-side (zero rows in $\sigma \mathcal{Y}^{\mathrm{T}} A_2 r_2(x)$) remain unaffected and $f(x) = 0$ is required for the rest of the rows in the equation. In fact, $f(x) = 0$ sketches the equilibrium manifold of the fast reaction system, as a quasi-steady-state solution that should be included in the reformulated system. In sum, the reformulated DAE system is shown as below:

$$\mathcal{Y}_a^{\mathrm{T}} \dot{x} = \mathcal{Y}_a^{\mathrm{T}} A_1 r_1(x) + \mathcal{Y}_a^{\mathrm{T}} g(t), \tag{4.15a}$$

$$f(x) = 0, \tag{4.15b}$$

$$\mathcal{Z}^{\mathrm{T}} \dot{x} = \mathcal{Z}^{\mathrm{T}} A_1 r_1(x) + \mathcal{Z}^{\mathrm{T}} g(t). \tag{4.15c}$$

**Reformulated Propoxylation Reactor Model**

Applying the nullspace projection method, the reformulated model can be derived after a sequence of matrix operations. For detailed information please refer to Appendix B. As a result of the reformulation procedure, two pseudo-species X and Y are introduced:

$$\begin{aligned} \mathrm{X}_n &= \mathrm{G}_n + \mathrm{D}_n, \qquad n = 0, 1, \ldots, N; \\ \mathrm{Y}_n &= \mathrm{Q}_n + \mathrm{R}_n, \qquad n = 0, 1, \ldots, N. \end{aligned} \tag{4.16}$$

In fact, X refers to the polyol product and Y the unsaturated byproduct. Following the reformulated system shown in Eqs. (4.15a)-(4.15c), the nullspace multiplication (Eq. (4.15c)) leads to a group of population balances that are not affected by the exchange reactions, which can be written in the form of concentrations:

$$\frac{\mathrm{d}(V[\mathrm{X_0}])}{\mathrm{d}t} = V(2k_h[\mathrm{W}] - k_i[\mathrm{G_0}])[\mathrm{M}], \tag{4.17a}$$
$$\frac{\mathrm{d}(V[\mathrm{X_1}])}{\mathrm{d}t} = V(k_i[\mathrm{G_0}] - k_p[\mathrm{G_1}])[\mathrm{M}], \tag{4.17b}$$
$$\frac{\mathrm{d}(V[\mathrm{X_n}])}{\mathrm{d}t} = Vk_p([\mathrm{G_{n-1}}] - [\mathrm{G_n}])[\mathrm{M}], \qquad n = 2, \ldots, N-1, \tag{4.17c}$$
$$\frac{\mathrm{d}(V[\mathrm{X_N}])}{\mathrm{d}t} = Vk_p[\mathrm{G_{N-1}}][\mathrm{M}]; \tag{4.17d}$$
$$\frac{\mathrm{d}(V[\mathrm{Y_0}])}{\mathrm{d}t} = -Vk_i[\mathrm{Q_0}][\mathrm{M}] + Vk_t \sum_{n=0}^{N}([\mathrm{G_n}] + [\mathrm{Q_n}])[\mathrm{M}], \tag{4.17e}$$
$$\frac{\mathrm{d}(V[\mathrm{Y_1}])}{\mathrm{d}t} = V(k_i[\mathrm{Q_0}] - k_p[\mathrm{Q_1}])[\mathrm{M}], \tag{4.17f}$$
$$\frac{\mathrm{d}(V[\mathrm{Y_n}])}{\mathrm{d}t} = Vk_p([\mathrm{Q_{n-1}}] - [\mathrm{Q_n}])[\mathrm{M}], \qquad n = 2, \ldots, N-1, \tag{4.17g}$$
$$\frac{\mathrm{d}(V[\mathrm{Y_N}])}{\mathrm{d}t} = Vk_p[\mathrm{Q_{N-1}}][\mathrm{M}]. \tag{4.17h}$$

In Eq. (4.13a), the range space term $\mathcal{Y}^{\mathrm{T}}A_2$ has two zero rows, corresponding to the balance equations of the water and monomer, and Eq. (4.15a) renders the same equations as stated in Eqs. (4.1a) and (4.2). The quasi-steady-state manifolds are obtained by using Eq. (4.15b):

$$[\mathrm{G_n}] \sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) = [\mathrm{D_n}] \sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]), \qquad n = 0, 1, \ldots, N; \tag{4.18a}$$
$$[\mathrm{Q_n}] \sum_{m=0}^{N}([\mathrm{D_m}] + [\mathrm{R_m}]) = [\mathrm{R_n}] \sum_{m=0}^{N}([\mathrm{G_m}] + [\mathrm{Q_m}]), \qquad n = 0, 1, \ldots, N. \tag{4.18b}$$

To this end, the reformulation procedure is complete with Eqs. (4.16), (4.17) and (4.18). Furthermore, the total amount of catalyst is equal to the amount of chains with $\mathrm{K^+}$ ions,

denoted as:

$$n_c = V \sum_{n=0}^{N} ([\mathrm{G_n}] + [\mathrm{Q_n}]); \tag{4.19}$$

and additionally, we introduce $n_i$ as the total number of moles of the initiator:

$$n_i = V \sum_{n=0}^{N} ([\mathrm{G_n}] + [\mathrm{D_n}]); \tag{4.20}$$

and the total moles of the unsaturated chains:

$$n_u = V \sum_{n=0}^{N} ([\mathrm{Q_n}] + [\mathrm{R_n}]). \tag{4.21}$$

As a result, Eqs. (4.16) and (4.18) can be further reduced to:

$$\begin{aligned} \mathrm{X}_n n_c &= \mathrm{G}_n(n_i + n_u), \qquad n = 0, 1, \ldots, N; \\ \mathrm{Y}_n n_c &= \mathrm{Q}_n(n_i + n_u), \qquad n = 0, 1, \ldots, N. \end{aligned} \tag{4.22}$$

By using the definition of $n_c$, the right hand side of Eq. (4.17e) can also be simplified because $n_c$ is constant for most applications. In sum, the reformulated model consists of

| | |
|---|---|
| Population balances | Eq. (4.17) |
| Quasi-steady states | Eq. (4.22) and definitions in Eqs. (4.19)- (4.21) |
| Additional equations | Monomer balance (Eq. (4.2)) |
| | Volume determination (Eqs. (4.3) and (4.4)) |
| | VLE relations (Eqs. (4.5)- (4.9)) |

Note that although the original stiff differential systems can be numerically handled by the Gear type methods, the nullspace projection procedure is a better solution. It gives an open equation system for optimization purposes. Also, the reformulated model is superior to the earlier one because it eliminates the fast dynamic modes in the differential equations, such that the DAE system becomes less stiff with a reduced number of differential equations.

## 4.3 Recipe Optimization Formulation

The recipe for a batch/semi-batch operation is often designed *off-line* based on the experience of past production runs. For many industrial processes, process recipe improvement is only carried out manually based on laboratory experiments and process simulation programs. However, given the dynamic model developed above, more rigorous model-based optimization methods can be exploited, providing more insight of the process and accurate calculation results. In this study, the optimization problem is formulated to minimize the batch time by designing the optimum reactor temperature and monomer feeding profiles. The constraints on the process deal with the final product quality and process safety regulations, including the target product molecular weight as well as thresholds on byproduct formation, unreacted monomer, reactor temperature, etc.

In the polyol industry, a number of quantities have been widely used to characterize the product performance, such as the molecular weight, level of unsaturated monofunctional chains (termed as unsat number in the remainder), functionality, hydroxyl number, and polydispersity index. These indices are greatly influenced by the choice of the starting alcohol and initial charge condition and are also subject to the variation of operating conditions. All these quality indices can be readily calculated given the type of the starter used and the population distribution of the polyol. In this work, we consider the target number average molecular weight and unsat number as key quality requirements that the optimized recipe should satisfy. To calculate the number average molecular weight, we introduce the notation of polymer moments. By definition, the $k$th moment of a polymer species (e.g. $\mathrm{X}$) is written as

$$\gamma_k = \sum_{n=1}^{N} n^k \mathrm{X}_n, \quad k = 0, 1, \ldots, \tag{4.23}$$

and the number average molecular weight can then be defined further as the ratio of the

first and zeroth moment, multiplied by the molecular weight of the repeating unit:

$$M_n = MW_{\mathrm{PO}} \frac{\gamma_1}{\gamma_0}. \tag{4.24}$$

Note that the ratio of the first and zeroth moment also represents the number average chain length of the polymer. Similarly, for the weight average molecular weight $M_w$, we define

$$M_w = MW_{\mathrm{PO}} \frac{\gamma_2}{\gamma_1}. \tag{4.25}$$

Moreover, the ratio of the $M_w$ over $M_n$ defines $\mathrm{PDI}$, which is an index that accounts for the spread of the molecular distribution:

$$\mathrm{PDI} = \frac{M_w}{M_n}. \tag{4.26}$$

The unsat number calibrates the concentration of monofunctional chains in the final product. It is defined as the milliequivalents of unsat chains per total mass:

$$\mathrm{unsat} = 1000 \frac{n_u}{m}. \tag{4.27}$$

The monofunctionality is undesired because it generally decreases the functionality that in consequence strongly affects the viscosity of the product; thus, corresponding upper bounds should be enforced with regard to specific product categories and applications. Also, the concentration of unreacted monomer in the final polyol product should be maintained under proper limits. Conventionally, it is measured in parts per million (ppm), as shown below:

$$\mathrm{unrct} = MW_{\mathrm{PO}} \frac{\mathrm{M}}{m} \times 10^6. \tag{4.28}$$

Functionality is defined as the ratio of the total amount of hydroxyl groups from the

initiator and monol over the total amount of all types of polymer chains:

$$f = \frac{N_{OH} n_i + n_u}{n_i + n_u}. \tag{4.29}$$

In the definition, $N_{OH}$ corresponds to the number of branches of the initial alcohol molecule ($N_{OH} = 2$ for propylene glycol). And also, the hydroxyl equivalent weight (HEW) is defined as the number-average molecular weight divided by the functionality:

$$\text{HEW} = \frac{M_n}{f}. \tag{4.30}$$

Lastly, the number of hydroxyl groups (OH#) gives the hydroxyl content of a polyol, calculated by using the equivalent weight of KOH:

$$\text{OH\#} = \frac{1000 MW_{\text{KOH}}}{\text{HEW}}. \tag{4.31}$$

In addition, process safety is always a vital concern in the polymerization reaction process. To derive the safety constraints, we first write a simplified energy balance equation of the reactor by only considering the propagation reaction as the source of reaction heat:

$$\frac{\mathrm{d}(mH_b)}{\mathrm{d}t} = F \Delta H_f MW_{\text{PO}} + r_p(-\Delta H_p) MW_{\text{PO}} - q, \tag{4.32}$$

where $r_p$ is the lumped rate of all propagation reactions, and the heat of reaction $-\Delta H_p$ is assumed to be constant. Also, $q$ denotes the heat removal rate from the heat exchanger that can be determined by using the overall heat transfer coefficient $U$ and area $A$:

$$q = UA(T - T_w), \tag{4.33}$$

where $T_w$ is the temperature of the water used by the heat exchanger, and it is assumed to be constant. In Eq. (4.32), the enthalpies of the feed flow and bulk liquid are $H_f$ and $H_b$,

respectively, which are defined as:

$$H_i = \int c_{pi} \mathrm{d}T, \quad i = \{f, b\}. \tag{4.34}$$

The heat capacity of the feed monomer $c_{pf}$ is cubic with respect to temperature, and the bulk heat capacity $c_{pb}$ can be estimated by the heat capacity of the product polyol, which is almost linear with temperature.

For the first safety constraint, the heat removal duty cannot exceed the allowed maximum cooling capacity of the heat exchanger attached to the reactor:

$$r_p(-\Delta H_p) MW_{\mathrm{PO}} \leqslant F(-\Delta H_f) MW_{\mathrm{PO}} + UA(T - T_w). \tag{4.35}$$

Here, we assume the monomer feed enters at a constant temperature $T_m = 25^{\circ}\mathrm{C}$, which is lower than the reactor temperature, offering extra cooling capability in addition to the heat exchanger capacity. And the term $UA$ represents the heat transfer efficiency that is known as a constant. Secondly, for the polyol process, the amount of unreacted oxides present in the reactor should be carefully controlled to prevent the plant from risks of product decomposition, under the accidental circumstance that the plant loses its cooling capability during operations. To carry out such a task, it is conducive to add a constraint on the *adiabatic end temperature* [146], which equals the summation of the current reactor temperature and the potential adiabatic temperature rise due to the occurrence of *total loss of cooling*. When the heat exchanger breaks down at time $t_c$ with the reactor temperature noted as $T_c$, it is clear that $q = 0$ and also reasonable to assume $F = 0$ after $t_c$. Therefore, integrating Eq. (4.32) starting from $t_c$ to the steady state (infinity) gives:

$$m(H_b(T_{ad}) - H_b(T_c)) = V[\mathrm{M}] MW_{\mathrm{PO}}(-\Delta H_p), \tag{4.36}$$

where, $T_{ad}$ is the adiabatic end temperature and we assume all the monomers are con-

sumed in the propagation reactions. For the safety limit of the adiabatic end temperature, a sufficient safety margin is also needed to tolerate uncertainties, where a recommended value of $250°\mathrm{C}$ is reported in a patent document [147].

The process recipe is optimized to shorten the operating time demanded for polymerization and the optimization model consists of the reformulated reactor model and additional process constraints; this leads to a large set of differential and algebraic equations. The problem can be written in a general form as noted in Eq. (2.2), with the objective function defined as the total time length.

## 4.4 Case Study

In the case study, the process of interest is the production of a low molecular weight polypropylene glycol ($M_n = 950\ \mathrm{g/mol}$) from the polymerization of $\mathrm{PO}$ initiated by water and propylene glycol ($\mathrm{PG}$). The product is widely used in applications such as coating and surfactants. For this example, the basic ingredients are given as follow:

| | |
|---|---|
| Starter: | PG and Water, |
| Catalyst: | KOH, |
| Monomer: | PO. |

### 4.4.1 Model Implementation

The polymerization model is implemented in GAMS after discretization into a NLP. The size of the model is largely determined by the discretization setting and the number of the recorded chain length limit. A careful choice is required to balance the accuracy and computational load of the model. In this study, we assume the controls are represented by piecewise linear functions that preserve continuity over finite elements, and twenty-four equidistant finite elements are used along with three Radau collocation points in

the orthogonal collocation scheme. The chain length distribution is truncated at $N = 18$, which is large enough to accommodate the long chains.

Initially, we test the fidelity of the developed first-principles model. To validate the model against plant data, the participating model parameters such as kinetic constants and thermodynamic properties need to be adjusted such that the model can adequately represent the real process, despite the imposed assumptions made and process uncertainties. The adjustment actions are based on the understanding and experience with the process, aiming to deal with deficiencies of the original process model [148]. Next, optimization is performed over the verified model, but allowing for variations of the reactor temperature and feed rate in certain ranges.

### 4.4.2 Model Validation

In this study, the reactor pressure profile is employed as the main criterion for tuning to illustrate the validation procedure. In fact, a much more detailed validation with polymer quality indices has been done, but proprietary considerations prevent us from presenting more information here. To obtain the pressure, the VLE calculations from Eqs. (4.5)-(4.9) are essential but further complicated in the presence of nitrogen and a vent system control valve in the reactor.

First, the nitrogen inside the reactor tank contributes to the total pressure. As the polymerization takes place, the nitrogen partial pressure rises over time since the liquid volume expands and compresses the gases. Assume the initial amount of nitrogen $n_{\mathrm{N}_2}$ is known, then the real-time partial pressure over $\mathrm{N}_2$ can be obtained by the ideal gas law, since the reactor pressure is not extremely high:

$$P_{\mathrm{N}_2}\bar{V} = n_{\mathrm{N}_2}RT. \tag{4.37}$$

Here, the gas phase volume $\bar{V}$ equals the total reactor volume minus the liquid phase volume $V$. Note that the total reactor pressure in Eq. (4.6) also includes the partial pressure of nitrogen. Next, the installed control valve avoids extreme pressures in the reactor tank: once the total pressure exceeds the upper limit $P^{max}$, the valve opens and keeps the pressure below $P^{max}$. This operation does not affect the reactions in the liquid phase, if we assume a negligible loss of the vapor phase $\mathrm{PO}$ when the system is vented. In addition, as nitrogen escapes, $n_{\mathrm{N_2}}$ decreases in time. The amount of released nitrogen can be estimated using Eq. (4.37) as the gas phase volume decreases and the reactor pressure stays constant at the maximum allowed by the vent system. A base recipe from real plant data is used for model calibration, where the batch time is normalized to unity, and the reactor temperature, monomer feeding rate, and pressure are recorded during each sampling interval. For this particular example, we consider the model prediction and real plant pressure profiles under the same operating conditions, and the result is shown in Fig. 4.1 in comparison. The model exhibits satisfactory performance in pressure prediction, and the discrepancy after valve relief is still acceptable, given the simplification made for gas release. The estimated partial pressures are also depicted. Note that for the last $6\%$ of the operation time, the corresponding plant data are not plotted due to full reactor venting. The major parameter adjustments are regarding reaction kinetics: all the kinetic constants are first obtained from published articles; next the pre-exponential factors are tuned to best fit the model predicted pressure to the plant data. Please refer to Appendix B for details.

### 4.4.3 Recipe Optimization Results

Optimization is carried out with respect to the same polymerization system, with the following constraints added:

1. The final number average MW of the polymer is no less than $950\ \mathrm{g/mol}$;

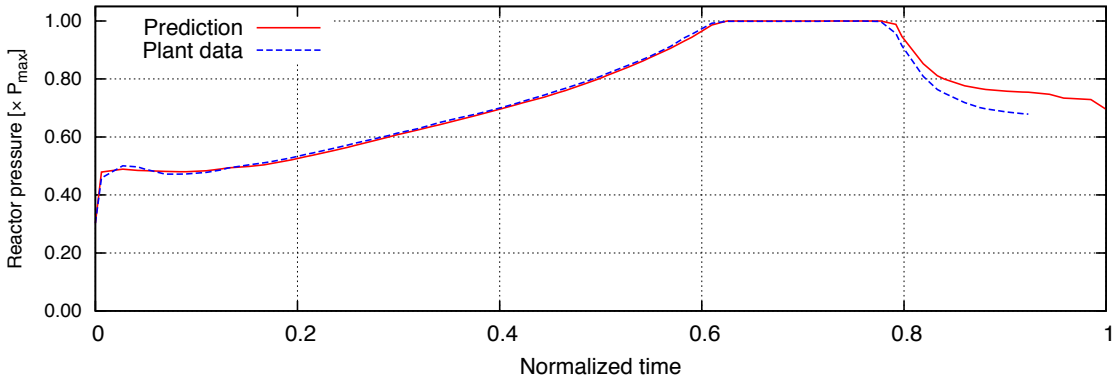



Figure 4.1: Reactor pressure profiles

2. The maximum unsaturation value is $0.033\,\mathrm{mmol/g\,polyol}$;

3. The final unreacted PO is no higher than $120\ \mathrm{ppm}$;

4. The maximum heat removal duty of the heat exchange is $UA(T - T_w)$;

5. The upper limit of the *adiabatic end temperature* is $(T_b + 80)^{\circ}\mathrm{C}$;

Here, the threshold values on product quality are obtained from simulating the base case recipe, where the polymerization model is solved with the reaction time and controls fixed to their recipe values. Among them, $UA$ and $T_b$ are constant parameters that are specified according to the reactor configuration. The optimization problem is solved with GAMS/IPOPT to proved local optimality with appropriate initialization of the participating variables, and all computations are performed on a laptop with a quad-core $2.80$ GHz Intel®i7 processor and $6$ GB memory, installed Linux kernel 3.2.0-14. Details on the statistics and solution of the model are tabulated in Tab. 4.2. The model can be solved within reasonable CPU time in minutes. The optimization model is initialized by using the simulation result of the base case recipe. The optimal solution renders a batch processing time of $0.53$ (normalized time), which is $47\%$ less than the base case recipe. Meanwhile, the quality constraints on the product are satisfied at the end of the operation. It should be noted that the base case recipe and set of process constraints are chosen to illustrate the use of dynamic optimization and do not necessarily reflect the true capa-

| Opt. soln | MW (g/mol) | Unsat (mmol/g) | PO (ppm) | # of var. | # of con. | CPU(s) |
|---|---|---|---|---|---|---|
| 0.53 | 950 | 0.033 | 120 | 10946 | 11043 | 56 |

Table 4.2: Recipe optimization model statistics and results

bility or restrictions of the plant. Nevertheless, the actual potential reaction time saving may vary, but it is still significant.

The obtained optimal operating strategy is depicted in Fig. 4.2. The optimized controls are shown in solid lines in comparison with the plant recipe in dashed lines. In the optimized scheme, the reactor temperature exhibits a U-shaped pattern: a high reactor temperature at the beginning period of operation can accelerate the hydrolysis and initiation reaction and therefore better *kick off* the following polymerization. And after that, the reactor temperature plunges down and stays low (lower than the corresponding recipe value) due to the process safety and product quality constraints. However, after the feeding period is over, the reactor temperature rises up for quick monomer digestion in the last few percent of the operation time horizon. The feed rate profile also starts at a high level, and gradually decreases during most of the operation time period. An important difference between the current plant recipe and the optimized one is that there are obviously two periods in the plant recipe: PO feeding followed by PO digestion. On the contrary, the optimized recipe tends to merge the two periods, and therefore the rising temperature at the end is needed to achieve the desired final level of PO. In Fig. 4.3, we show the transition behavior of two critical constraining factors: the adiabatic end temperature and the heat removal rate. The adiabatic end temperature constraint is not active, since it does not reach the upper bound specified by $T_{ad} - T_b = 80\,^{\circ}\mathrm{C}$. The cooling capacity is the major limiting factor of the process, which stays active for most of the operation time horizon. The total capacity limit also includes a portion provided by the monomer feed. Note that here we assume the heat exchanger is not fouled, otherwise the optimal solution may drift. Considering heat exchanger fouling can bring additional complexity to the optimization

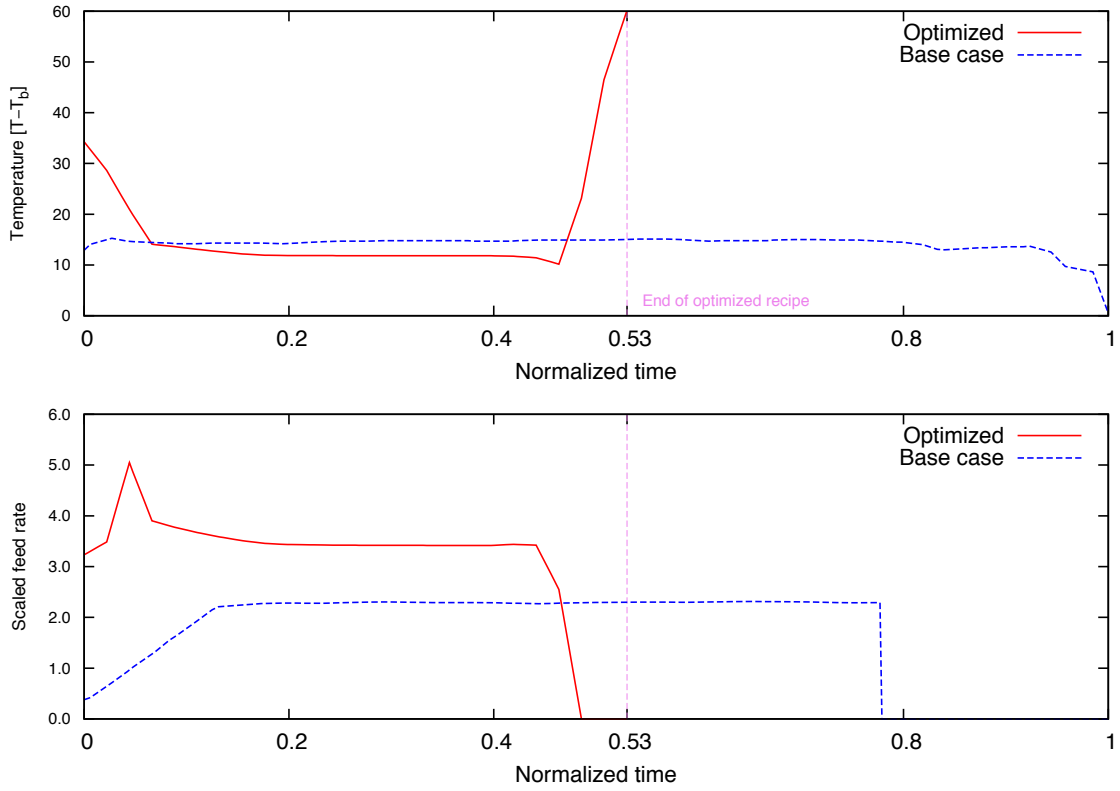



Figure 4.2: Optimal control profiles of the process

problem [134], and it remains an interesting future extension to our current work.

Fig. 4.4 shows the population growth of the product and unsat polymer chains, where a number of species of particular chain lengths is presented. Note that the upper limit of $18$ repeating units is admissible since the mole number of the longest chain stays close to zero for both types of polymers. During the polymerization, the monomers are continually added to the polymers and chains of higher lengths gradually appear and grow. A product chain of length $n$ can either become length $n + 1$ through the propagation reaction or transform into a dormant chain of equal length. It is also worth noting that the populations of the unsat chains sharply increase in the digestion period because of the rising reactor temperature. The final time population distributions of the two polymers are given in histograms shown in Fig. 4.5. The product polymer nearly follows the *Poisson* distribution, which agrees with the fact that the main population is nearly a well-defined living system. The maximum in the population is located at $n = 7$, which is also the

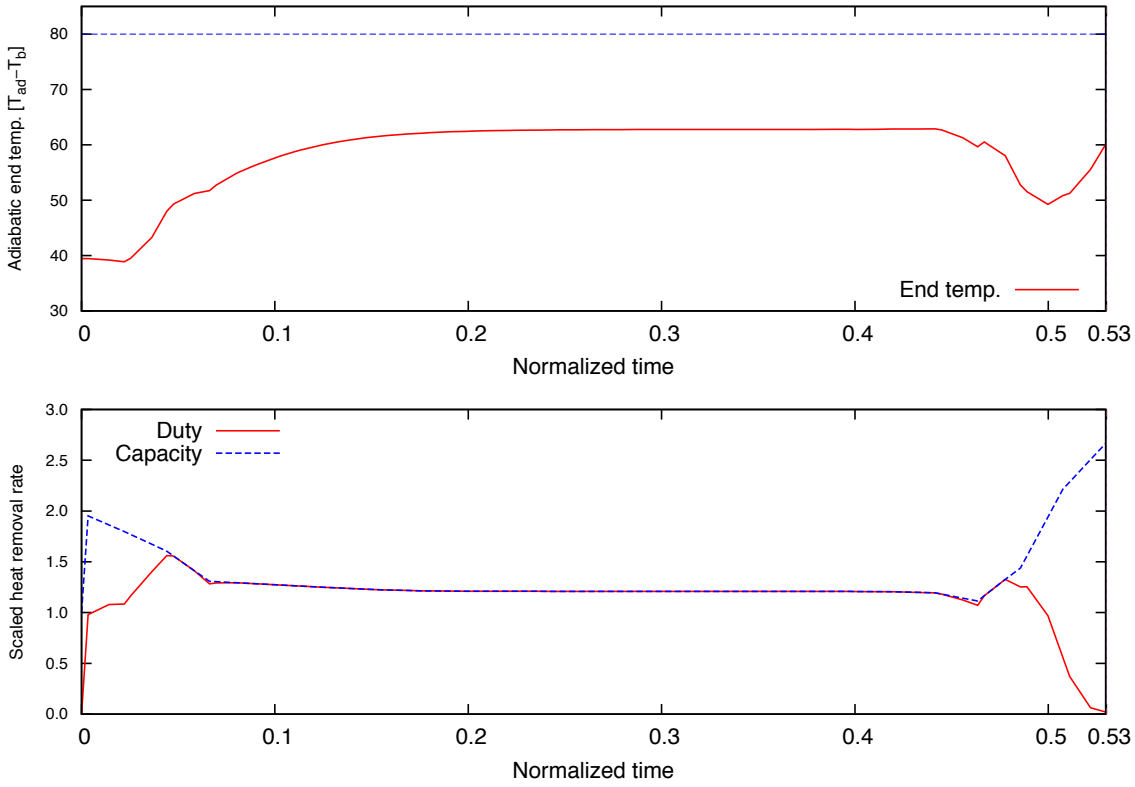



Figure 4.3: Process constraint profiles

central position of the distribution. However, the distribution of the unsat chains significantly differs from the product, where the majority has less than 12 repeating units and the distribution follows in a descending manner from short chains to long ones, except for the initiation chains $n = 0$, where the initiation reaction is faster than propagation. The MWD information is valuable for analyzing the product polyol properties, such as viscosity.

Fig. 4.6 demonstrates the optimized number-average molecular weights and the PDI for both the product and the unsat (in solid lines), in comparison with the profiles obtained from the base case (in dashed lines). For the product chains, the optimized recipe reaches the same MW and PDI as the base case. But for the unsats, the PDI is higher for the optimized recipe while the MWs are very close in both cases. Note that the base case profiles show a time period in the end that has no significant changes in all quantities, but after optimization, those properties still change until the end of the batch. The number-average

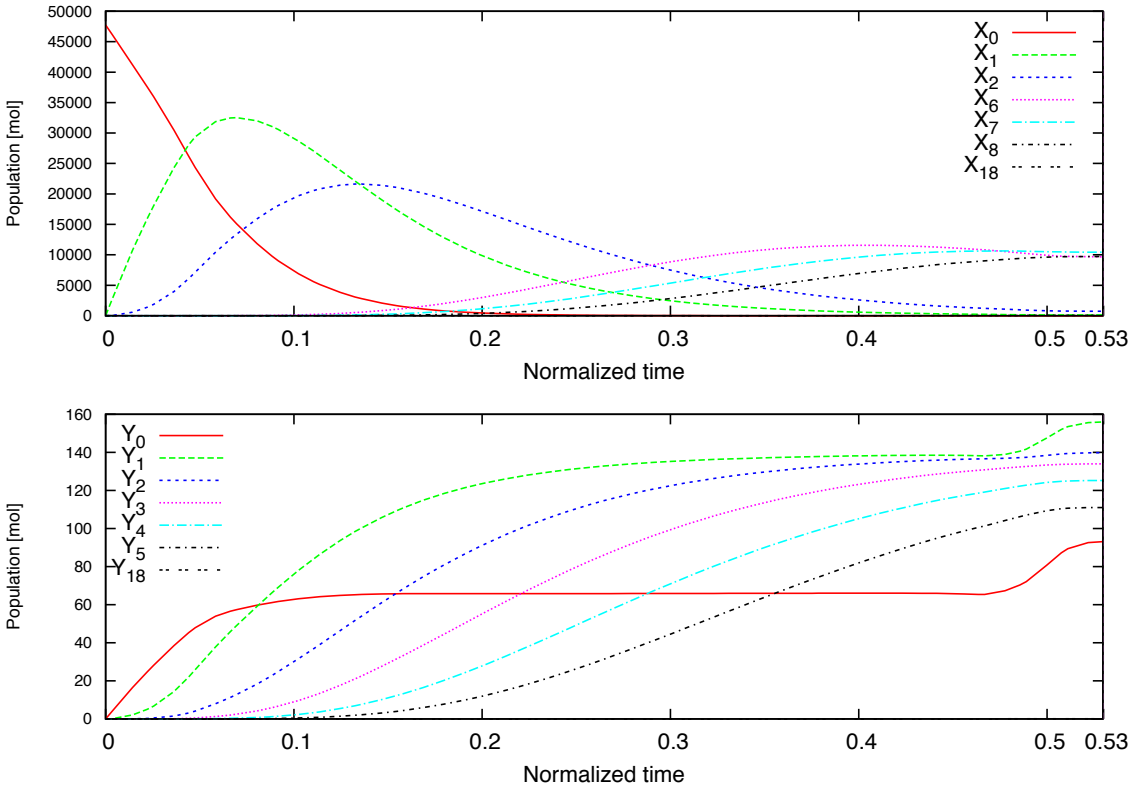



Figure 4.4: Population growth profiles of polymers of different chain lengths

MW of the unsat chains is of the same order of magnitude of the product polymers. A large unsat number is particularly undesirable in this case, since it implies a considerable amount of monomer is consumed by the byproduct. In addition, the unsat chains have a larger PDI which indicates their distribution is more widely spread, agreeing with the distributions shown in Fig. 4.5. Lastly, a group of commonly used quality indices are depicted in Fig. 4.7. It can be concluded that their final-time values are in proper ranges.

## 4.5 Concluding Remarks

In this chapter, we have addressed the reactor modeling and dynamic optimization of polyether polyol processes, using the production of polypropylene glycol as the example. The first-principle reactor model has been established through applying conservation

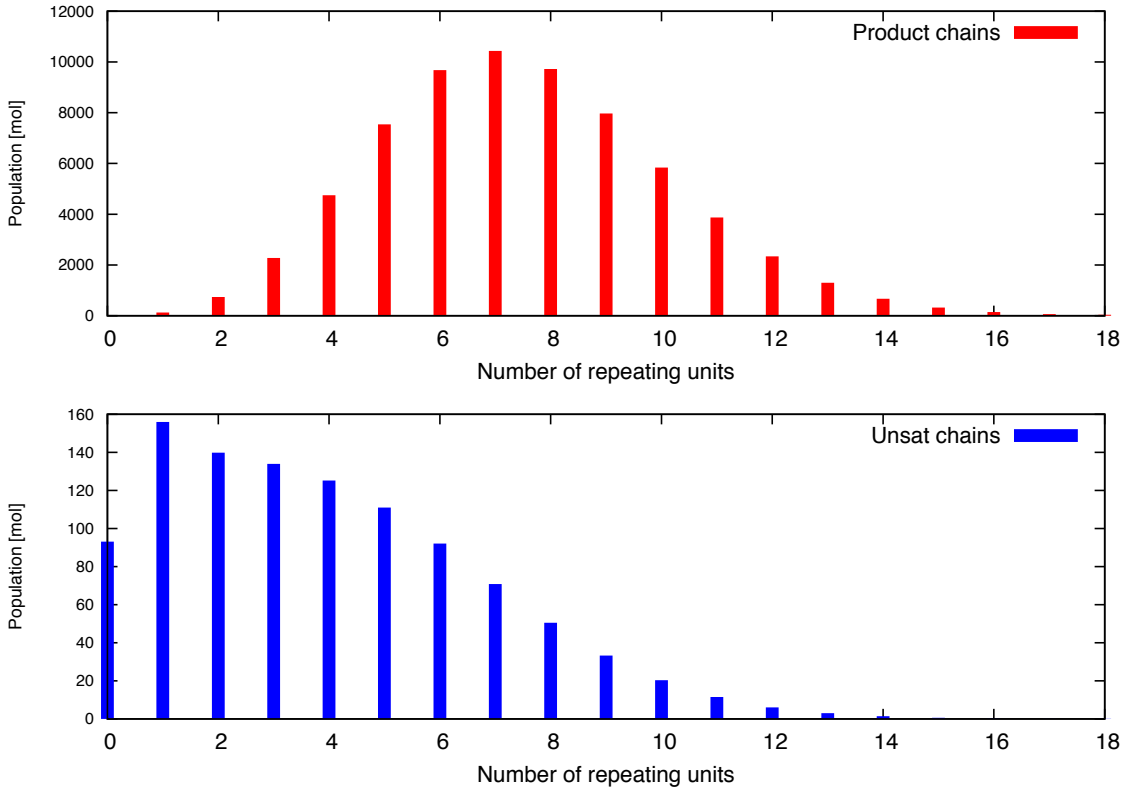



Figure 4.5: Molecular weight distributions in the final product

laws, reaction kinetic relations, etc. and the original model demonstrated two-time-scale dynamic behaviors because of the presence of the fast cation exchange reactions. Therefore, a reformulation procedure was conducted by using the nullspace projection method, which aimed at separating the fast dynamic modes and modeling them as algebraic equations with regard to the quasi-steady states. The established model was validated against plant data using the reactor pressure profile. A number of key kinetic parameters from published literature has been adjusted during the model calibration process. Next, a dynamic optimization problem was formulated to improve the polymerization recipe design by minimizing the batch time. Several important limiting factors were introduced as the constraints in the optimization problem, concerning the product quality and process safety. The study results illustrate detailed information on the dynamic characteristics of the polymerization process and show very promising performance of the optimized recipe by significantly reducing the required polymerization time. Particularly for the

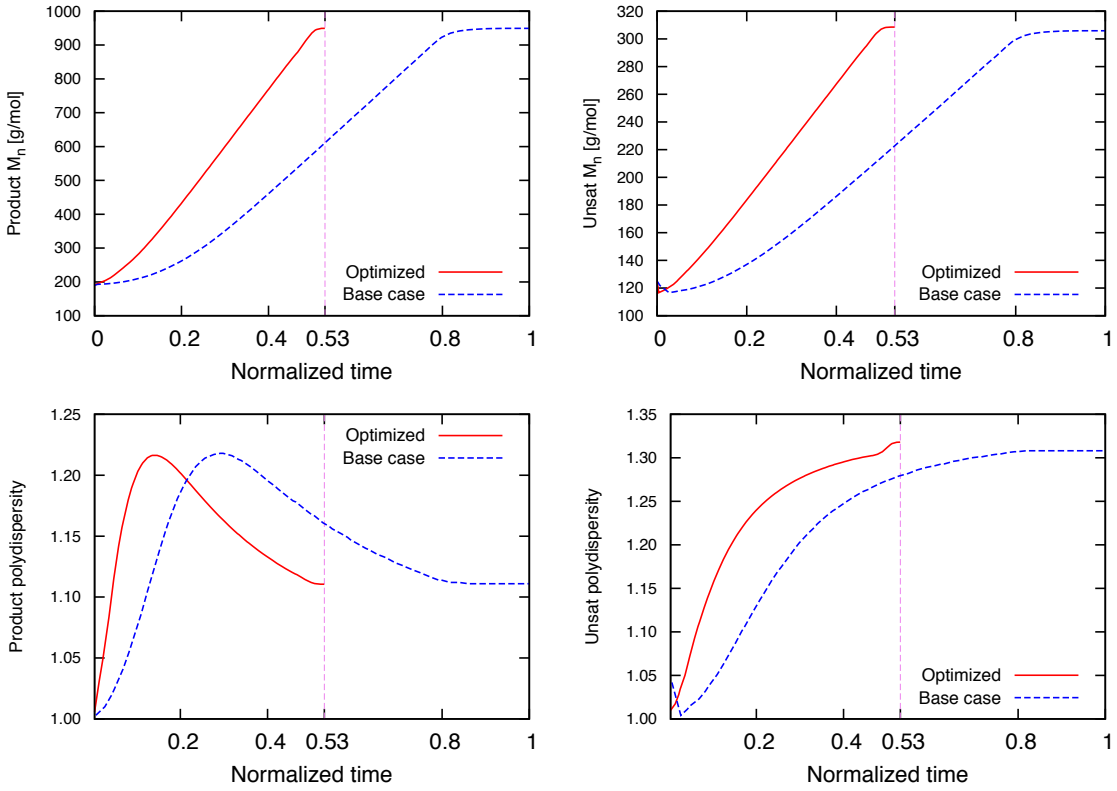



Figure 4.6: Polymer property profiles: molecular weights and polydispersity indices

control design, the optimizer showed the trend to merge the feeding and digestion periods, which has changed the design pattern used for the base case recipe.

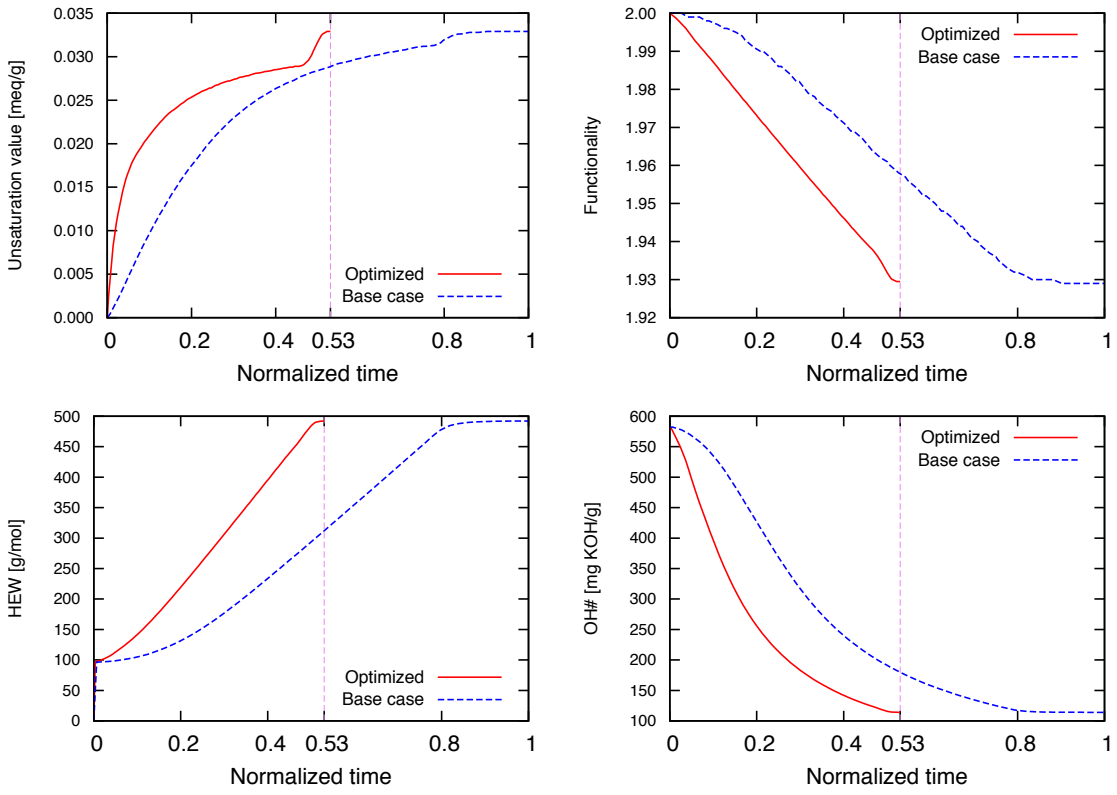



Figure 4.7: Polymer property profiles: important quality indices of polyols

## 4.6 Reactor Model Notation

| | |
|---|---|
| $a$ | liquid phase activity, dimensionless |
| $A$ | heat transfer area, $\mathrm{m}^2$ |
| $A_r$ | pre-exponential factor, $\mathrm{m}^3/\mathrm{mol} \cdot \mathrm{s}$ |
| $c_p$ | specific heat capacity, $\mathrm{J/g} \cdot \mathrm{K}$ |
| D | dormant product chains |
| $E_r$ | activation energy, $\mathrm{J/mol}$ |
| $f$ | functionality, dimensionless |
| $F$ | monomer feed rate, $\mathrm{mol/s}$ |
| G | growing product chains |
| $H$ | enthalpy, $\mathrm{J/g}$ |
| $\Delta H$ | reaction heat, $\mathrm{J/g}$ |
| $k$ | reaction rate constant, $\mathrm{m}^3/\mathrm{mol} \cdot \mathrm{s}$ |
| $l$ | number average chain length, dimensionless |
| $m$ | total mass, $\mathrm{g}$ |
| M | monomer |
| $M_n$ | number-average molecular weight of polyol, $\mathrm{g/mol}$ |
| $M_w$ | weight-average molecular weight of polyol, $\mathrm{g/mol}$ |
| $MW$ | molecular weight, $\mathrm{g/mol}$ |
| $n$ | number of moles, $\mathrm{mol}$ |
| $N$ | maximum number of repeating units |
| $P$ | pressure, $\mathrm{kPa}$ |
| $q$ | heat removal rate, $\mathrm{J/s}$ |
| Q | growing unsat chains |
| $r$ | reaction rate, $\mathrm{mol/s}$ |
| R | dormant unsat chains |
| $R$ | universal gas constant, $8.314\ \mathrm{J/mol} \cdot \mathrm{K}$ |
| $t$ | time, $\mathrm{s}$ |
| $T$ | temperature, $\mathrm{K}$ |
| $T_{ad}$ | adiabatic end temperature, $\mathrm{K}$ |
| $T_b$ | reference temperature, $\mathrm{K}$ |
| $U$ | overall heat transfer coefficient, $\mathrm{W/m}^2 \cdot \mathrm{K}$ |
| $V$ | liquid volume, $\mathrm{m}^3$ |
| $\bar{V}$ | gas volume, $\mathrm{m}^3$ |
| W | water |
| X | total product chains |
| Y | total unsat chains |
| | |
| $\gamma$ | polymer moment, $\mathrm{mol}$ |

| | |
|---|---|
| $\phi$ | lattice fraction, dimensionless |
| $\chi$ | interaction parameter, dimensionless |
| | |
| $[\cdot]$ | concentration, $\mathrm{mol/m^3}$ |

| *Subscripts* | |
|---|---|
| Substances | |
| $b$ | bulk |
| $c$ | catalyst |
| $f$ | feed |
| $i$ | initiator |
| $m$ | repeating units |
| $n$ | repeating units |
| $p$ | polymer |
| $s$ | solvent |
| $u$ | unsaturated chains |
| $w$ | water |
| Reactions | |
| $h$ | hydrolysis |
| $i$ | initiation |
| $p$ | propagation |
| $e$ | exchange |
| $t$ | transfer |

# Chapter 5

# Reactor Model Development and Copolymerization Recipe Optimization

This chapter continues our study on the modeling and recipe optimization of polymerization reactors. We extend the homopolymerization model of polyether polyols to a broader class that is termed as ring-opening polymerization processes. Also, the polymerization mechanism is complicated by the addition of copolymers. Moreover, in addition to the population balance method, we also develop the reactor model by using the method of moments. Both reactor models are tested to match their predictions with historical plant data. Again, we solve recipe optimization problems to minimize the polymerization time. In the case study example, the moment model shows superiority over the population balance model in terms of computational efficiency.

## 5.1 Background Information

A great variety of cyclic monomers have been successfully polymerized by the ring-opening polymerization process [149]. Many commercial polymers produced this way

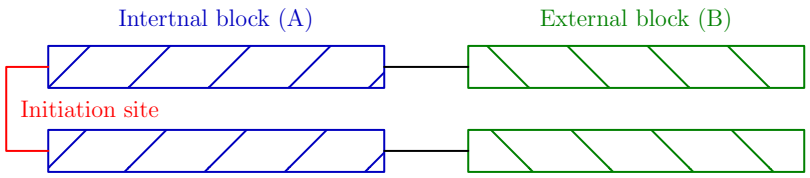



Figure 5.1: General structure of a block copolymer

are statistical or block copolymers with molecular weights (MWs) from a few hundred to several million. Conventionally, a block copolymer can be synthesized by a semi-batch polymerization process by successive feeds of respective monomers [141]. For instance, a block copolymer made from monomer A and B is shown in 5.1 with a cascade structure. The initiator has two branches, and each branch consists of a linear internal A block and external B block in tandem. The external block may contain a small amount of A in practice. Modeling such a polymerization process requires sufficient knowledge of the reaction kinetics, and process safety regulations are also important considerations for process optimization [150]. In this example, monomer B is potentially explosive such that the reactor for making this polymer needs to be blanketed with nitrogen at the start.

Population balance equations are often used in macro-scale polymerization reactor modeling, which correspond to a set of differential mole balance equations that can reveal the evolution of the complete chain length distribution over time. The size of the resulting balance equation system is decided by the breadth of the polymer chain length distribution, where modeling high MW polymers can lead to a considerably large-scale model that is computationally demanding. On the contrary, the method of moments [151], as a classical modeling approach, represents the average polymer properties based on statistical quantities, namely, moments. The moments can be applied to derive commonly used quality indices such as number/weight average molecular weight, polydispersity index (PDI), and monomer consumption/polymer production rates. The size of moment models is usually small and is not influenced by the chain length of polymers. As a result, solving a moment model requires much less computational effort than the full population

balance model. In this study, we derive both population balance and moment models; the population balance model is appropriate for process recipe simulation, while the moment model facilitates recipe design optimization.

## 5.2 Reactor Model Development

To produce monomer blocks, monomer A is first fed into the reactor continuously to grow the internal block. Next, monomer B is allowed to enter the reactor to form the external block. A degassing step can be performed to eliminate the unreacted A by vacuum distillation before feeding B to obtain a purer external block. In the first stage of polymerizing A, four primary reactions are considered: the initiation, propagation, transfer and exchange reactions, similar to the case for polyether polyols. In the secondary step, B is added to form terminal blocks by the corresponding initiation and propagation steps with higher reactivity than A. However, B does not participate in transfer reactions. If A is not completely digested or degassed, then A and B coexist in the reactor in the second stage. Under this circumstance, there are four propagation reactions, shown as follows:

| Propagation scheme | Rate constant |
|---|---|
| $\ldots \mathrm{A}^* + \mathrm{A} \rightarrow \ldots \mathrm{A}^*$ | $k_p^{AA}$ |
| $\ldots \mathrm{A}^* + \mathrm{B} \rightarrow \ldots \mathrm{B}^*$ | $k_p^{AB}$ |
| $\ldots \mathrm{B}^* + \mathrm{A} \rightarrow \ldots \mathrm{A}^*$ | $k_p^{BA}$ |
| $\ldots \mathrm{B}^* + \mathrm{B} \rightarrow \ldots \mathrm{B}^*$ | $k_p^{BB}$ |

Terminal unit is designated by $*$

Two associated reactivity ratios are defined as $r^A = k_p^{AA}/k_p^{AB}$ and $r^B = k_p^{BB}/k_p^{BA}$.

Depending on the different functional end groups, we define:

$G_{n_A,n_B}$ to denote the growing product chains;

$D_{n_A,n_B}$ to denote the dormant product chains;

$Q_{n_A,n_B}$ to denote the growing byproduct chains;

$R_{n_A,n_B}$ to denote the dormant byproduct chains.

Moreover, superscripts are introduced to the polymer species to indicate the terminal repeating unit or monomer type, except for the initiators that do not have any repeating units ($n_A = n_B = 0$). The reaction scheme is summarized in Tab. 5.1.

The byproduct chains share the same set of kinetic parameters as the product chains in chain initiation, growth, exchange and transfer. The four propagation rates are non-identical.

Initiation:

$$\mathrm{G}_{0,0} + \mathrm{M}^{\mathrm{A}} \xrightarrow{k_{\mathrm{i}}^{\mathrm{A}}} \mathrm{G}_{1,0}^{\mathrm{A}}$$
$$\mathrm{Q}_{0,0} + \mathrm{M}^{\mathrm{A}} \xrightarrow{k_{\mathrm{i}}^{\mathrm{A}}} \mathrm{Q}_{1,0}^{\mathrm{A}}$$
$$\mathrm{G}_{0,0} + \mathrm{M}^{\mathrm{B}} \xrightarrow{k_{\mathrm{i}}^{\mathrm{B}}} \mathrm{G}_{0,1}^{\mathrm{B}}$$
$$\mathrm{Q}_{0,0} + \mathrm{M}^{\mathrm{B}} \xrightarrow{k_{\mathrm{i}}^{\mathrm{B}}} \mathrm{Q}_{0,1}^{\mathrm{B}}$$

Propagation:

$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{M}^{\mathrm{A}} \xrightarrow{k_{\mathrm{p}}^{\mathrm{AA}}} \mathrm{G}_{\mathrm{n_A}+1,\mathrm{n_B}}^{\mathrm{A}} \quad (\mathrm{n_A}+\mathrm{n_B} \geqslant 1)$$
$$\mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{M}^{\mathrm{A}} \xrightarrow{k_{\mathrm{p}}^{\mathrm{AA}}} \mathrm{Q}_{\mathrm{n_A}+1,\mathrm{n_B}}^{\mathrm{A}} \quad (\mathrm{n_A}+\mathrm{n_B} \geqslant 1)$$
$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{M}^{\mathrm{B}} \xrightarrow{k_{\mathrm{p}}^{\mathrm{AB}}} \mathrm{G}_{\mathrm{n_A},\mathrm{n_B}+1}^{\mathrm{B}} \quad (\mathrm{n_A}+\mathrm{n_B} \geqslant 1)$$
$$\mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{M}^{\mathrm{B}} \xrightarrow{k_{\mathrm{p}}^{\mathrm{AB}}} \mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}+1}^{\mathrm{B}} \quad (\mathrm{n_A}+\mathrm{n_B} \geqslant 1)$$
$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{M}^{\mathrm{A}} \xrightarrow{k_{\mathrm{p}}^{\mathrm{BA}}} \mathrm{G}_{\mathrm{n_A}+1,\mathrm{n_B}}^{\mathrm{A}} \quad (\mathrm{n_A}+\mathrm{n_B} \geqslant 1)$$
$$\mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{M}^{\mathrm{A}} \xrightarrow{k_{\mathrm{p}}^{\mathrm{BA}}} \mathrm{Q}_{\mathrm{n_A}+1,\mathrm{n_B}}^{\mathrm{A}} \quad (\mathrm{n_A}+\mathrm{n_B} \geqslant 1)$$
$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{M}^{\mathrm{B}} \xrightarrow{k_{\mathrm{p}}^{\mathrm{BB}}} \mathrm{G}_{\mathrm{n_A},\mathrm{n_B}+1}^{\mathrm{B}} \quad (\mathrm{n_A}+\mathrm{n_B} \geqslant 1)$$
$$\mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{M}^{\mathrm{B}} \xrightarrow{k_{\mathrm{p}}^{\mathrm{BB}}} \mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}+1}^{\mathrm{B}} \quad (\mathrm{n_A}+\mathrm{n_B} \geqslant 1)$$

Transfer:

$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{M}^{\mathrm{A}} \xrightarrow{k_{\mathrm{t}}} \mathrm{D}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{Q}_{0,0} \quad (\mathrm{n_A},\mathrm{n_B} \geqslant 0)$$
$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{M}^{\mathrm{A}} \xrightarrow{k_{\mathrm{t}}} \mathrm{D}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{Q}_{0,0} \quad (\mathrm{n_A},\mathrm{n_B} \geqslant 0)$$
$$\mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{M}^{\mathrm{A}} \xrightarrow{k_{\mathrm{t}}} \mathrm{R}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{Q}_{0,0} \quad (\mathrm{n_A},\mathrm{n_B} \geqslant 0)$$
$$\mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{M}^{\mathrm{A}} \xrightarrow{k_{\mathrm{t}}} \mathrm{R}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{Q}_{0,0} \quad (\mathrm{n_A},\mathrm{n_B} \geqslant 0)$$

Exchange:

$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{D}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{A}} \xrightarrow{k_e} \mathrm{D}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{G}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{A}} \quad (\mathrm{n_A},\mathrm{n_B},\mathrm{m_A},\mathrm{m_B} \geqslant 0)$$
$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{D}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \underset{k_e}{\overset{k_e}{\rightleftharpoons}} \mathrm{D}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{G}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \quad (\mathrm{n_A},\mathrm{n_B},\mathrm{m_A},\mathrm{m_B} \geqslant 0)$$
$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{D}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \xrightarrow{k_e} \mathrm{D}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{G}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \quad (\mathrm{n_A},\mathrm{n_B},\mathrm{m_A},\mathrm{m_B} \geqslant 0)$$
$$\mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{R}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{A}} \xrightarrow{k_e} \mathrm{R}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{Q}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{A}} \quad (\mathrm{n_A},\mathrm{n_B},\mathrm{m_A},\mathrm{m_B} \geqslant 0)$$
$$\mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{R}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \underset{k_e}{\overset{k_e}{\rightleftharpoons}} \mathrm{R}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{Q}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \quad (\mathrm{n_A},\mathrm{n_B},\mathrm{m_A},\mathrm{m_B} \geqslant 0)$$
$$\mathrm{Q}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{R}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \xrightarrow{k_e} \mathrm{R}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{Q}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \quad (\mathrm{n_A},\mathrm{n_B},\mathrm{m_A},\mathrm{m_B} \geqslant 0)$$
$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{R}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{A}} \underset{k_e}{\overset{k_e}{\rightleftharpoons}} \mathrm{D}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{Q}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{A}} \quad (\mathrm{n_A},\mathrm{n_B},\mathrm{m_A},\mathrm{m_B} \geqslant 0)$$
$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{R}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \underset{k_e}{\overset{k_e}{\rightleftharpoons}} \mathrm{D}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{A}} + \mathrm{Q}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \quad (\mathrm{n_A},\mathrm{n_B},\mathrm{m_A},\mathrm{m_B} \geqslant 0)$$
$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{R}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{A}} \underset{k_e}{\overset{k_e}{\rightleftharpoons}} \mathrm{D}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{Q}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{A}} \quad (\mathrm{n_A},\mathrm{n_B},\mathrm{m_A},\mathrm{m_B} \geqslant 0)$$
$$\mathrm{G}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{R}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \underset{k_e}{\overset{k_e}{\rightleftharpoons}} \mathrm{D}_{\mathrm{n_A},\mathrm{n_B}}^{\mathrm{B}} + \mathrm{Q}_{\mathrm{m_A},\mathrm{m_B}}^{\mathrm{B}} \quad (\mathrm{n_A},\mathrm{n_B},\mathrm{m_A},\mathrm{m_B} \geqslant 0)$$

Table 5.1: Reactions in anionic ring-opening copolymerization

### 5.2.1 Population Balance Model Equations

The population balance equations for individual polymeric species can be established as a set of first-order ordinary differential equations:

$$\begin{aligned}\frac{\mathrm{d}G_{0,0}}{\mathrm{d}t} =& V^{-1}[-(k_i^A M^A + k_i^B M^B)G_{0,0} - k_t G_{0,0} M^A - k_e G_{0,0} \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B}) \\ &+ k_e D_{0,0} \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})],\end{aligned} \tag{5.1a}$$

$$\begin{aligned}\frac{\mathrm{d}G_{1,0}^A}{\mathrm{d}t} =& V^{-1}[k_i^A G_{0,0} M^A - (k_p^{AA} M^A + k_p^{AB} M^B)G_{1,0}^A - k_t G_{1,0}^A M^A - k_e G_{1,0}^A \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B}) \\ &+ k_e D_{1,0}^A \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})],\end{aligned} \tag{5.1b}$$

$$\begin{aligned}\frac{\mathrm{d}G_{0,1}^B}{\mathrm{d}t} =& V^{-1}[k_i^B G_{0,0} M^B - (k_p^{BA} M^A + k_p^{BB} M^B)G_{0,1}^B - k_t G_{0,1}^B M^A - k_e G_{0,1}^B \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B}) \\ &+ k_e D_{0,1}^B \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})],\end{aligned} \tag{5.1c}$$

$$\begin{aligned}\frac{\mathrm{d}G_{n_A,n_B}^A}{\mathrm{d}t} =& V^{-1}[(k_p^{AA} G_{n_A-1,n_B}^A + k_p^{BA} G_{n_A-1,n_B}^B)M^A - (k_p^{AA} M^A + k_p^{AB} M^B)G_{n_A,n_B}^A - k_t G_{n_A,n_B}^A M^A \\ &- k_e G_{n_A,n_B}^A \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B}) + k_e D_{n_A,n_B}^A \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})], \\ & n_A \geqslant 2, n_B \geqslant 0,\end{aligned} \tag{5.1d}$$

$$\begin{aligned}\frac{\mathrm{d}G_{n_A,n_B}^B}{\mathrm{d}t} =& V^{-1}[(k_p^{AB} G_{n_A,n_B-1}^A + k_p^{BB} G_{n_A,n_B-1}^B)M^B - (k_p^{BA} M^A + k_p^{BB} M^B)G_{n_A,n_B}^B - k_t G_{n_A,n_B}^B M^A \\ &- k_e G_{n_A,n_B}^B \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B}) + k_e D_{n_A,n_B}^B \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})], \\ & n_A \geqslant 0, n_B \geqslant 2,\end{aligned} \tag{5.1e}$$

$$\begin{aligned}\frac{\mathrm{d}D_{n_A,n_B}^S}{\mathrm{d}t} =& V^{-1}[k_t G_{n_A,n_B}^S M^A + k_e G_{n_A,n_B}^S \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B}) - k_e D_{n_A,n_B}^S \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})], \\ & S \in \{A, B\}, n_A \geqslant 0, n_B \geqslant 0;\end{aligned} \tag{5.1f}$$

$$\frac{\mathrm{d}Q_{0,0}}{\mathrm{d}t} = V^{-1}[-(k_i^B M^B + k_i^A M^A)Q_{0,0} + k_t \sum_{n_A=0}^{\infty}\sum_{n_B=0}^{\infty}(G_{n_A,n_B} + Q_{n_A,n_B})M^A - k_t Q_{0,0} M^A$$
$$- k_e Q_{0,0} \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B}) + k_e R_{0,0} \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})], \tag{5.1g}$$

$$\frac{\mathrm{d}Q_{1,0}^A}{\mathrm{d}t} = V^{-1}[k_i^A Q_{0,0} M^A - (k_p^{AA} M^A + k_p^{AB} M^B)Q_{1,0}^A - k_t Q_{1,0}^A M^A - k_e Q_{1,0}^A \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B})$$
$$+ k_e R_{1,0}^A \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})], \tag{5.1h}$$

$$\frac{\mathrm{d}Q_{0,1}^B}{\mathrm{d}t} = V^{-1}[k_i^B Q_{0,0} M^B - (k_p^{BA} M^A + k_p^{BB} M^B)Q_{0,1}^B - k_t Q_{0,1}^B M^A - k_e Q_{0,1}^B \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B})$$
$$+ k_e R_{0,1}^B \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})], \tag{5.1i}$$

$$\frac{\mathrm{d}Q_{n_A,n_B}^A}{\mathrm{d}t} = V^{-1}[(k_p^{AA} Q_{n_A-1,n_B}^A + k_p^{BA} Q_{n_A-1,n_B}^B)M^A - (k_p^{AA} M^A + k_p^{AB} M^B)Q_{n_A,n_B}^A - k_t Q_{n_A,n_B}^A M^A$$
$$- k_e Q_{n_A,n_B}^A \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B}) + k_e R_{n_A,n_B}^A \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})],$$
$$n_A \geqslant 2, n_B \geqslant 0, \tag{5.1j}$$

$$\frac{\mathrm{d}Q_{n_A,n_B}^B}{\mathrm{d}t} = V^{-1}[(k_p^{AB} Q_{n_A,n_B-1}^A + k_p^{BB} Q_{n_A,n_B-1}^B)M^B - (k_p^{BA} M^A + k_p^{BB} M^B)Q_{n_A,n_B}^B - k_t Q_{n_A,n_B}^B M^A$$
$$- k_e Q_{n_A,n_B}^B \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B}) + k_e R_{n_A,n_B}^B \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})],$$
$$n_A \geqslant 0, n_B \geqslant 2, \tag{5.1k}$$

$$\frac{\mathrm{d}R_{n_A,n_B}^S}{\mathrm{d}t} = V^{-1}[k_t Q_{n_A,n_B}^S M^A + k_e Q_{n_A,n_B}^S \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(D_{m_A,m_B} + R_{m_A,m_B}) - k_e R_{n_A,n_B}^S \sum_{m_A=0}^{\infty}\sum_{m_B=0}^{\infty}(G_{m_A,m_B} + Q_{m_A,m_B})],$$
$$S \in \{A, B\}, n_A \geqslant 0, n_B \geqslant 0. \tag{5.1l}$$

Here the polymer species without a superscript index refer to the total population, regardless of the different terminal units:

$$x_{n_A,n_B} = \sum_{S\in\{A,B\}} x_{n_A,n_B}^S, \quad x \in \{G, D, Q, R\}. \tag{5.2}$$

The monomer balance equations are defined for both monomers. Monomer A is fed into the reactor at rate $F^A$, and consumed in the initiation, propagation, and chain transfer reactions resulting in byproduct chains. The balance equation is similar for monomer B, except for the different initiation and propagation rates and absence of the proton transfer reaction term.

$$\begin{aligned}\frac{\mathrm{d}M^A}{\mathrm{d}t} =& F^A - V^{-1}[k_i^A(G_{0,0}+Q_{0,0}) + k_p^{AA}\sum_{n_A=1}^{\infty}\sum_{n_B=0}^{\infty}(G^A_{n_A,n_B}+Q^A_{n_A,n_B}) \\ &+ k_p^{BA}\sum_{n_A=0}^{\infty}\sum_{n_B=1}^{\infty}(G^B_{n_A,n_B}+Q^B_{n_A,n_B}) + k_t\sum_{n_A=0}^{\infty}\sum_{n_B=0}^{\infty}(G_{n_A,n_B}+Q_{n_A,n_B})]M^A,\end{aligned} \tag{5.3a}$$

$$\begin{aligned}\frac{\mathrm{d}M^B}{\mathrm{d}t} =& F^B - V^{-1}[k_i^B(G_{0,0}+Q_{0,0}) + k_p^{AB}\sum_{n_A=1}^{\infty}\sum_{n_B=0}^{\infty}(G^A_{n_A,n_B}+Q^A_{n_A,n_B}) \\ &+ k_p^{BB}\sum_{n_A=0}^{\infty}\sum_{n_B=1}^{\infty}(G^B_{n_A,n_B}+Q^B_{n_A,n_B})]M^B.\end{aligned} \tag{5.3b}$$

For the total mass balance, as the monomers consecutively enter the polymerization system, we have:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = F^A MW_A + F^B MW_B, \tag{5.4}$$

where $MW_A$ and $MW_B$ denote the molecular weights of the two monomers, respectively. The liquid volume is calculated by:

$$V = m[10^{-6} + 7.576\times10^{-10}(T-298.15)]. \tag{5.5}$$

The total reactor pressure $P$ is calculated as the sum of the partial pressures of volatile components, determined from the liquid phase activities and saturated vapor pressures:

$$P = \sum_i P_i, \tag{5.6a}$$

$$P_i = a_i P_i^{sat}. \tag{5.6b}$$

The vapor pressures by the Antoine equation:

$$\log_{10} P_i^{sat} = A_i - \frac{B_i}{T + C_i}. \tag{5.7}$$

The activities $a_i$ are given by the Flory-Huggins theory. In the polymerization process, possible volatile components include the monomer A, B, initiators, etc. As a result, multiple components exist in the liquid and gaseous phase simultaneously. However, as a rational method to deal with polymer-solvent equilibrium, the Flory-Huggins theory is in principle applicable to binary mixtures of a polymer and solvent. To manage the VLE calculation, we make the following assumptions. First, the polymer chains are totally in the liquid phase; secondly the polymer solution can be viewed as a ternary mixture where A and B (denoted in index $s_A$ and $s_B$ respectively) are two types of solvents and all the other species are treated as polymer (denoted in index $p$); lastly, the interaction between the two solvent molecules is ignored. Hence, the activities of A and B are determined via:

$$\ln a_{s_A} = \ln \phi_{s_A} + (1 - \frac{1}{l})\phi_A + \chi^A \phi_A^2; \tag{5.8a}$$

$$\ln a_{s_B} = \ln \phi_{s_B} + (1 - \frac{1}{l})\phi_B + \chi^B \phi_B^2. \tag{5.8b}$$

In Eq. (5.8), the lattice fractions $\phi_{s_A}$, $\phi_{s_B}$ and $\phi_p$ are used instead of mole fractions. The interaction parameters $\chi^A$ and $\chi^B$ are non-dimensional. Here the interaction effect of monomer A and B molecules is not considered for simplicity as well as the fact that a successive feeding pattern is used for block polymers. A more rigorous treatment is proposed by Favre et al. [152] at the cost of additional complexity in activity calculation with more model parameters. The number average chain length $l$ is taken into account when

calculating the lattice fraction:

$$\phi_{s_A} = \frac{n_{s_A}}{n_{s_A} + n_{s_B} + n_p l}; \tag{5.9a}$$

$$\phi_{s_B} = \frac{n_{s_B}}{n_{s_A} + n_{s_B} + n_p l}; \tag{5.9b}$$

$$\phi_p = \frac{n_p l}{n_{s_A} + n_{s_B} + n_p l}. \tag{5.9c}$$

Here, $n_{s_A}$, $n_{s_B}$ and $n_p$ are the numbers of moles of the solvents and polymer. On the other hand, the activities of other volatile components can be treated as constants for simplicity, and their influence on the total reactor pressure is often found to be minor. This simplified VLE model is valid because the dominating liquid components are monomer A, B and polymer for the majority of time.

The reactor model developed above consists of population, monomer and total mass balances, as well as volume and VLE calculations. The size of the model becomes considerably large when modeling high MW polymers, since the detailed chain length distribution is calculated.

### 5.2.2 Reformulated Reactor Model

We apply the nullspace projection method to systematically reformulate the reaction equation. Two pseudo-species X and Y are introduced:

$$\begin{aligned} X^S_{n_A,n_B} &= G^S_{n_A,n_B} + D^S_{n_A,n_B}, \quad S \in \{A, B\}, n_A, n_B \geqslant 0; \\ Y^S_{n_A,n_B} &= Q^S_{n_A,n_B} + R^S_{n_A,n_B}, \quad S \in \{A, B\}, n_A, n_B \geqslant 0. \end{aligned} \tag{5.10}$$

We introduce the following notation: first, the total amount of catalyst is equal to the amount of chains attached with base metal ions, denoted as

$$n_c = \sum_{n_A=0}^{\infty} \sum_{n_B=0}^{\infty} (G_{n_A,n_B} + Q_{n_A,n_B}); \tag{5.11}$$

and additionally, we introduce $n_i$ as the total number of moles of the initiator

$$n_i = \sum_{n_A=0}^{\infty} \sum_{n_B=0}^{\infty} (G_{n_A,n_B} + D_{n_A,n_B}); \tag{5.12}$$

and the total moles of the byproduct chains

$$n_u = \sum_{n_A=0}^{\infty} \sum_{n_B=0}^{\infty} (Q_{n_A,n_B} + R_{n_A,n_B}). \tag{5.13}$$

After reformulation, the differential population balance equations for X and Y are expressed in Eqs. (5.14), as shown below:

$$\frac{\mathrm{d}X_{0,0}}{\mathrm{d}t} = V^{-1}(-k_i^B M^B - k_i^A M^A) G_{0,0}, \tag{5.14a}$$

$$\frac{\mathrm{d}X_{1,0}^A}{\mathrm{d}t} = V^{-1}[k_i^A G_{0,0} M^A - (k_p^{AA} M^A + k_p^{AB} M^B) G_{1,0}^A], \tag{5.14b}$$

$$\frac{\mathrm{d}X_{0,1}^B}{\mathrm{d}t} = V^{-1}[k_i^B G_{0,0} M^B - (k_p^{BA} M^A + k_p^{BB} M^B) G_{0,1}^B], \tag{5.14c}$$

$$\frac{\mathrm{d}X_{1,1}^A}{\mathrm{d}t} = V^{-1}[k_p^{BA} G_{0,1}^B M^A - (k_p^{AA} M^A + k_p^{AB} M^B) G_{1,1}^A], \tag{5.14d}$$

$$\frac{\mathrm{d}X_{1,1}^B}{\mathrm{d}t} = V^{-1}[k_p^{AB} G_{1,0}^A M^B - (k_p^{BA} M^A + k_p^{BB} M^B) G_{1,1}^B], \tag{5.14e}$$

$$\frac{\mathrm{d}X_{n_A,n_B}^A}{\mathrm{d}t} = V^{-1}[(k_p^{AA} G_{n_A-1,n_B}^A + k_p^{BA} G_{n_A-1,n_B}^B) M^A - (k_p^{AA} M^A + k_p^{AB} M^B) G_{n_A,n_B}^A], \quad n_A \geqslant 2, n_B \geqslant 0, \tag{5.14f}$$

$$\frac{\mathrm{d}X_{n_A,n_B}^B}{\mathrm{d}t} = V^{-1}[(k_p^{AB} G_{n_A,n_B-1}^A + k_p^{BB} G_{n_A,n_B-1}^B) M^B - (k_p^{BA} M^A + k_p^{BB} M^B) G_{n_A,n_B}^B], \quad n_A \geqslant 0, n_B \geqslant 2; \tag{5.14g}$$

$$\frac{\mathrm{d}Y_{0,0}}{\mathrm{d}t} = V^{-1}[-(k_i^B M^B + k_i^A M^A)Q_{0,0} + k_t n_c M^A], \tag{5.14h}$$

$$\frac{\mathrm{d}Y_{1,0}^A}{\mathrm{d}t} = V^{-1}[k_i^A Q_{0,0} M^A - (k_p^{AA} M^A + k_p^{AB} M^B)Q_{1,0}^A], \tag{5.14i}$$

$$\frac{\mathrm{d}Y_{0,1}^B}{\mathrm{d}t} = V^{-1}[k_i^B Q_{0,0} M^B - (k_p^{BA} M^A + k_p^{BB} M^B)Q_{0,1}^B], \tag{5.14j}$$

$$\frac{\mathrm{d}Y_{1,1}^A}{\mathrm{d}t} = V^{-1}[k_p^{BA} Q_{0,1}^B M^A - (k_p^{AA} M^A + k_p^{AB} M^B)Q_{1,1}^A], \tag{5.14k}$$

$$\frac{\mathrm{d}Y_{1,1}^B}{\mathrm{d}t} = V^{-1}[k_p^{AB} Q_{1,0}^A M^B - (k_p^{BA} M^A + k_p^{BB} M^B)Q_{1,1}^B], \tag{5.14l}$$

$$\frac{\mathrm{d}Y_{n_A,n_B}^A}{\mathrm{d}t} = V^{-1}[(k_p^{AA} Q_{n_A-1,n_B}^A + k_p^{BA} Q_{n_A-1,n_B}^B)M^A - (k_p^{AA} M^A + k_p^{AB} M^B)Q_{n_A,n_B}^A], \quad n_A \geqslant 2, n_B \geqslant 0, \tag{5.14m}$$

$$\frac{\mathrm{d}Y_{n_A,n_B}^B}{\mathrm{d}t} = V^{-1}[(k_p^{AB} Q_{n_A,n_B-1}^A + k_p^{BB} Q_{n_A,n_B-1}^B)M^B - (k_p^{BA} M^A + k_p^{BB} M^B)Q_{n_A,n_B}^B], \quad n_A \geqslant 0, n_B \geqslant 2. \tag{5.14n}$$

Meanwhile, the algebraic equations giving the quasi-steady states of the exchange reactions are shown as:

$$\begin{aligned} X_{n_A,n_B}^S n_c &= G_{n_A,n_B}^S (n_i + n_u), \quad S \in \{A, B\}, n_A, n_B \geqslant 0; \\ Y_{n_A,n_B}^S n_c &= Q_{n_A,n_B}^S (n_i + n_u), \quad S \in \{A, B\}, n_A, n_B \geqslant 0. \end{aligned} \tag{5.15}$$

In sum, the reformulated model comprises three major building blocks:

| | |
|---|---|
| Population balances | Eq. (5.14) |
| Quasi-steady states | Eq. (5.15) and definitions in Eqs. (5.11)- (5.13) |
| Additional equations | Monomer balance (Eq. (5.3)) |
| | Volume determination (Eqs. (5.4) and (5.5)) |
| | VLE relations (Eqs. (5.6)- (5.9)) |

In the context of copolymerization, appropriate upper bounds are required for both monomers. The reformulated copolymerization model is of the same order of magnitude of the size as the original one.

### 5.2.3 Moment Model

The method of moments is a very well-known method for solving polymerization systems with very large number of individual species. The moment equations are derived from aggregating population balances with different weights. The obtained moment model retains information for tracking average polymer properties. The method of moments can be applied well to linear polymers. The following notation is introduced to develop the moment model for the copolymerization process:

- $\zeta$ moment of growing product chains ($\mathrm{G}$);
- $\nu$ moment of growing byproduct chains ($\mathrm{Q}$);
- $\lambda$ moment of product chains ($\mathrm{X}$);
- $\mu$ moment of byproduct chains ($\mathrm{Y}$).

The $k$th moment of a polymer species (e.g. $\mathrm{G}$) is defined as

$$\zeta_k = \sum_{n=1}^{\infty} n^k G_n, \quad k = 0, 1, \ldots, \tag{5.16}$$

where, $n$ represents the total number of repeating units irrespective of monomer types, viz. $n = n_A + n_B$, and $G_n$ is the population of living product chains of length $n$. The rest of the moments can be defined analogously. In this study, the moment model for the copolymerization of A and B needs to account for the effect of the chain-ends on propagation. Therefore, superscripts $A$ and $B$ are also used for the moment notation to

designate ending units. In Eqs. (5.17), the moment balances are derived:

$$\frac{\mathrm{d}X_0}{\mathrm{d}t} = V^{-1}(-k_i^A G_0 M^A - k_i^B G_0 M^B), \tag{5.17a}$$

$$\frac{\mathrm{d}\lambda_0^A}{\mathrm{d}t} = V^{-1}(k_i^A G_0 M^A + k_p^{BA}\zeta_0^B M^A - k_p^{AB}\zeta_0^A M^B), \tag{5.17b}$$

$$\frac{\mathrm{d}\lambda_0^B}{\mathrm{d}t} = V^{-1}(k_i^B G_0 M^B + k_p^{AB}\zeta_0^A M^B - k_p^{BA}\zeta_0^B M^A), \tag{5.17c}$$

$$\frac{\mathrm{d}\lambda_k^A}{\mathrm{d}t} = V^{-1}(k_i^A G_0 M^A + k_p^{AA}\sum_{i=0}^{k-1}\binom{k}{i}\zeta_i^A M^A + k_p^{BA}\sum_{i=0}^{k}\binom{k}{i}\zeta_i^B M^A - k_p^{AB}\zeta_k^A M^B), \quad k = 1, 2, \ldots, \tag{5.17d}$$

$$\frac{\mathrm{d}\lambda_k^B}{\mathrm{d}t} = V^{-1}(k_i^B G_0 M^B + k_p^{BB}\sum_{i=0}^{k-1}\binom{k}{i}\zeta_i^B M^A + k_p^{AB}\sum_{i=0}^{k}\binom{k}{i}\zeta_i^A M^B - k_p^{BA}\zeta_k^B M^A), \quad k = 1, 2, \ldots; \tag{5.17e}$$

$$\frac{\mathrm{d}Y_0}{\mathrm{d}t} = V^{-1}(-k_i^A Q_0 M^A - k_i^B Q_0 M^B + k_t n_c M^A), \tag{5.17f}$$

$$\frac{\mathrm{d}\mu_0^A}{\mathrm{d}t} = V^{-1}(k_i^A Q_0 M^A + k_p^{BA}\nu_0^B M^A - k_p^{AB}\nu_0^A M^B), \tag{5.17g}$$

$$\frac{\mathrm{d}\mu_0^B}{\mathrm{d}t} = V^{-1}(k_i^B Q_0 M^B + k_p^{AB}\nu_0^A M^B - k_p^{BA}\nu_0^B M^A), \tag{5.17h}$$

$$\frac{\mathrm{d}\mu_k^A}{\mathrm{d}t} = V^{-1}(k_i^A Q_0 M^A + k_p^{AA}\sum_{i=0}^{k-1}\binom{k}{i}\nu_i^A M^A + k_p^{BA}\sum_{i=0}^{k}\binom{k}{i}\nu_i^B M^A - k_p^{AB}\nu_k^A M^B), \quad k = 1, 2, \ldots, \tag{5.17i}$$

$$\frac{\mathrm{d}\mu_k^B}{\mathrm{d}t} = V^{-1}(k_i^B Q_0 M^B + k_p^{BB}\sum_{i=0}^{k-1}\binom{k}{i}\nu_i^B M^A + k_p^{AB}\sum_{i=0}^{k}\binom{k}{i}\nu_i^A M^B - k_p^{BA}\nu_k^B M^A), \quad k = 1, 2, \ldots. \tag{5.17j}$$

In addition, the algebraic equations for the quasi-steady states of the exchange reactions are obtained:

$$X_0 n_c = G_0(n_i + n_u), \tag{5.18a}$$

$$Y_0 n_c = Q_0(n_i + n_u), \tag{5.18b}$$

$$\lambda_k^S n_c = \zeta_k^S(n_i + n_u), \quad S = \{A, B\}, k = 0, 1, \ldots, \tag{5.18c}$$

$$\mu_k^S n_c = \nu_k^S(n_i + n_u), \quad S = \{A, B\}, k = 0, 1, \ldots, \tag{5.18d}$$

and the definitions of the total amounts of the catalyst, initiator and byproduct are rewritten as follows:

$$n_c = G_0 + Q_0 + \sum_{S\in\{A,B\}} \zeta_0^S + \nu_0^S, \tag{5.19a}$$

$$n_i = X_0 + \sum_{S\in\{A,B\}} \lambda_0^S, \tag{5.19b}$$

$$n_u = Y_0 + \sum_{S\in\{A,B\}} \mu_0^S. \tag{5.19c}$$

Lastly, the monomer balance equations for A and B are shown as:

$$\frac{\mathrm{d}M^A}{\mathrm{d}t} = F^A - V^{-1}[k_i^A(G_0 + Q_0) + k_p^{AA}(\zeta_0^A + \nu_0^A) + k_p^{BA}(\zeta_0^B + \nu_0^B) + k_t n_c]M^A; \tag{5.20a}$$

$$\frac{\mathrm{d}M^B}{\mathrm{d}t} = F^B - V^{-1}[k_i^B(G_0 + Q_0) + k_p^{AB}(\zeta_0^A + \nu_0^A) + k_p^{BB}(\zeta_0^B + \nu_0^B)]M^B. \tag{5.20b}$$

The constructed moment model for copolymerization is capable of predicting typically used polymer property indices. This assembles a reactor model that is significantly smaller than the full reactor model based on species balances. Although the detailed chain length distribution information is lost, the moment model is well suited for applications that only require average polymer properties. This becomes especially useful when computation resources are constrained and computation times are critical, such as (*on-line*) process

optimization.

## 5.3 Process Recipe Optimization

Similar to the previous study, we show that the developed reactor models are able to adequately represent the polymerization process through adjustments of key kinetic and VLE parameters for a known set of plant data in the case study example. The polymer property indices are also similarly defined. First, the number average molecular weight $M_n$ and weight average molecular weight $M_w$ can be obtained with the moments:

$$M_n = \overline{MW}\frac{\lambda_1}{\lambda_0}; \tag{5.21a}$$

$$M_w = \overline{MW}\frac{\lambda_2}{\lambda_1}. \tag{5.21b}$$

The molecular weight of the repeating units is calculated as a weighted average of the molecular weights of A and B:

$$\overline{MW} = wMW_A + (1-w)MW_B; \tag{5.22a}$$

$$w = \frac{\int_{t_0}^{t} F^A(\tau)\mathrm{d}\tau - M^A(t)}{\int_{t_0}^{t}[F^A(\tau)+F^B(\tau)]\mathrm{d}\tau - M^A(t) - M^B(t)}. \tag{5.22b}$$

In Eq. (5.22b), $w$ represents the mole fraction of A in the polymer chains. This instantaneous copolymer composition can also be estimated by using the Mayo-Lewis equation [153], when the quasi-steady state assumption on A and B ended chains (roughly equal rates of disappearance and formation) is valid. PDI is used as a measurement of the heterogeneity of the polymer to characterize the spread of the polymer chain length distribution, calculated by $M_w/M_n$. Narrow distribution, corresponding to low PDI, is preferred in many applications. The byproduct chains created by the transfer reaction are

measured in milliequivalents per total mass:

$$\alpha = 1000\frac{n_u}{m}. \tag{5.23}$$

The percentage of polymer chains ending with monomer B in the external block is an important property for the block copolymer, which can be calculated from the zeroth moments:

$$\epsilon = \frac{\lambda_0^B + \mu_0^B}{\sum\limits_{S\in\{A,B\}} (\lambda_0^S + \mu_0^S)} \times 100\%. \tag{5.24}$$

Also, the unreacted monomer in the product mixture at final time should stay below proper upper limits:

$$\beta^S = MW_{\mathrm{S}}\frac{M^S}{m} \times 10^6, \quad S \in \{A, B\}. \tag{5.25}$$

The process safety constraint regarding the heat removal duty and adiabatic end temperature can be derived in a similar manner as in the homopolymerization case, the key equations are given as follows:

$$\frac{\mathrm{d}(mH_b)}{\mathrm{d}t} = \sum_{S\in\{A,B\}} [F^S \Delta H_f^S + r_p^S(-\Delta H_p^S)]MW_{\mathrm{S}} - q, \tag{5.26}$$

$$q = UA(T - T_w), \tag{5.27}$$

$$\sum_{S\in\{A,B\}} r_p^S(-\Delta H_p^S)MW_{\mathrm{S}} \leqslant \sum_{S\in\{A,B\}} F^S(-\Delta H_f^S)MW_{\mathrm{S}} + UA(T - T_w). \tag{5.28}$$

$$m[H_b(T_{ad}) - H_b(T_c)] = \sum_{S\in\{A,B\}} M^S(-\Delta H_p^S)MW_{\mathrm{S}}, \tag{5.29}$$

Meanwhile, monomer B is potentially explosive in nitrogen, and a linear limit extrapolation of the explosive region boundary is given by

$$y_B \leqslant 1.3865 - 0.001764T - 0.0003568P, \tag{5.30}$$

and $y_B$ represents the vapor phase concentration of B.

The recipe optimization formulation include all constraints derived above to minimize the polymerization time. Note that the time horizon is divided into two segments corresponding to each monomer feeding period.

## 5.4 Case Study

We test our modeling and optimization framework with an example study of two-thousand dalton A-B block copolymers. The internal A block is produced in tandem with the external B block without the degassing step; but there is a specified minimum percentage of B ended chains that needs to be met. A real-world process recipe obtained from plant data is used as the base case in the following study. We validate the population balance model (PBM) and moment model (MM) with the base case plant data, and recipe optimization is performed by using both models. In the solution procedure, the discretization settings for the two models are different: a finer mesh of forty finite elements along with a two-point Radau collocation is applied to MM for each stage, while a coarser one of ten finite elements with three-point Radau collocation roots is introduced for PBM due to its large model size. In order to record the entire chain length distribution of the PBM, the upper bounds for repeating units are specified as $N_A = 30$ and $N_B = 14$. It is worth noting that a lower order collocation is used in MM, in conjunction with a larger number of finite elements; this in general excels in handling path constraints, particularly for high index ones, in the optimization problem.

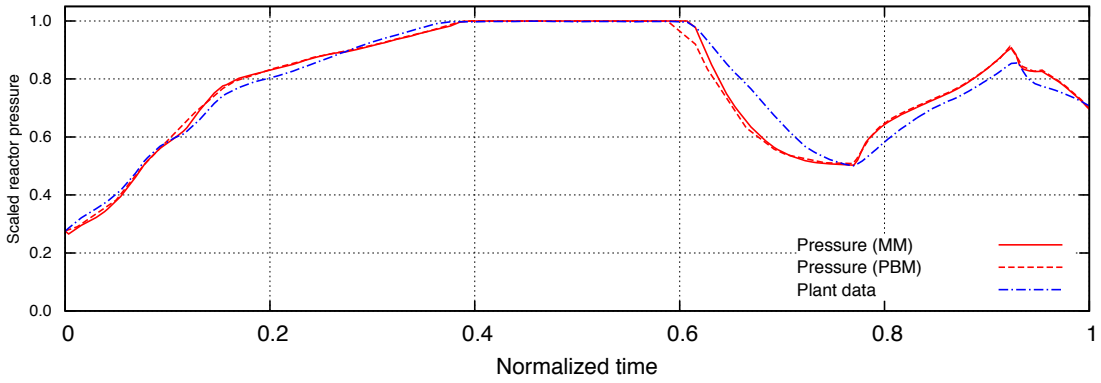



Figure 5.2: Reactor pressure profiles for model validation

### 5.4.1 Model Validation

We validate both reactor models with their pressure predictions against plant data under the same operating conditions, and the result is shown in Fig. 5.2. The match between the prediction curves and data trajectory is considered satisfactory. The same kinetics and VLE parameters are used in the PBM and MM.

### 5.4.2 Recipe Optimization Results

The optimization problems are solved with GAMS/IPOPT to proved local optimality with appropriate initialization of the participating variables; IPOPT runs with linear solver MA86. All computations are performed on a desktop with an $8$-core $2.80$ GHz Intel®i7 processor, $9$ GB memory, and installed Linux kernel 3.2.0-34.

#### Optimization with Moment Models

The moment model is well suited for recipe optimization owing to its small scale and ability to predict product properties. The optimized recipe is able to reduce the copolymerization processing time by $42.0\%$. Details on the statistics and solution of the model can be found in Tab. 5.2. The computation load is quite manageable for optimization with

| Opt. soln | # of var. | # of con. | CPU (s) |
|---|---|---|---|
| 0.580 | 11,248 | 11,121 | 22 |

Table 5.2: Recipe optimization statistics and results from the moment model

MM.

The operating strategies for the copolymerization process are depicted in Fig. 5.3. For the base case recipe, the polymerization temperature is designed to remain constant over time for the A and B addition stages. The recipe design pattern for the monomer feed policy is as follows: A and B are fed consecutively with a noticeable gap in between. Each feeding window begins with a *ramping* period, where the feed rate increases roughly linearly, and then the rate is kept at a desired constant level. The polymerization is continued with no monomer feed for digestion.

The optimized recipe redesigns the temperature and feed rate profiles. The reactor temperature rises sharply during the A digestion and B feeding period, reaching its upper limit. In the feed profile, B is gradually added after A, and the digestion period of B is almost negligible by virtue of its high reactivity. Fig. 5.4 shows the three important constraining factors of the process: the adiabatic end temperature, heat removal rate and B concentration in the vapor phase. The value of the adiabatic end temperature indicates the potential effect of the latent heat existing in unreacted monomers on the reactor temperature. This constraint is not active since the monomer inventories are well controlled. On the contrary, the reactor cooling capacity is the primary limiting factor for further improvement of the process performance and the corresponding constraint is active during most of the operation time horizon, especially for the A feeding stage. The total cooling capacity also includes a portion provided by the monomer feed streams that are of a lower temperature than the reactor. Finally, we inspect the safety constraint regarding the vapor phase composition distribution, and the system is found to be within the safety zone predicted by Eq. (5.30).

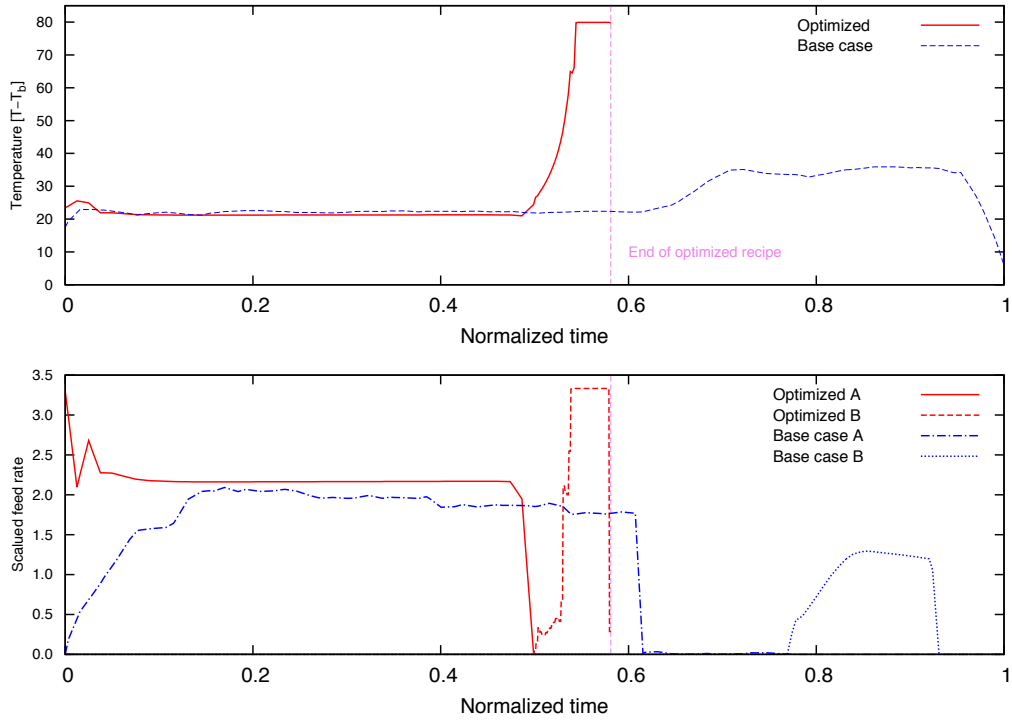



Figure 5.3: Optimal control profiles of the process from the moment model

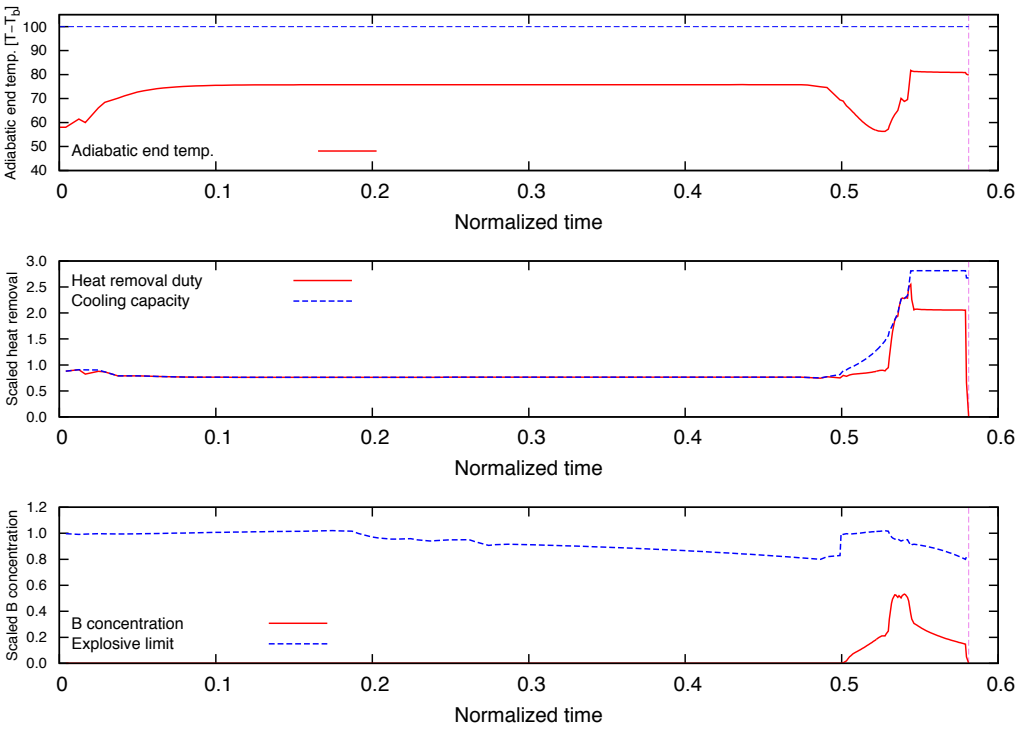



Figure 5.4: Process constraint profiles from the moment model

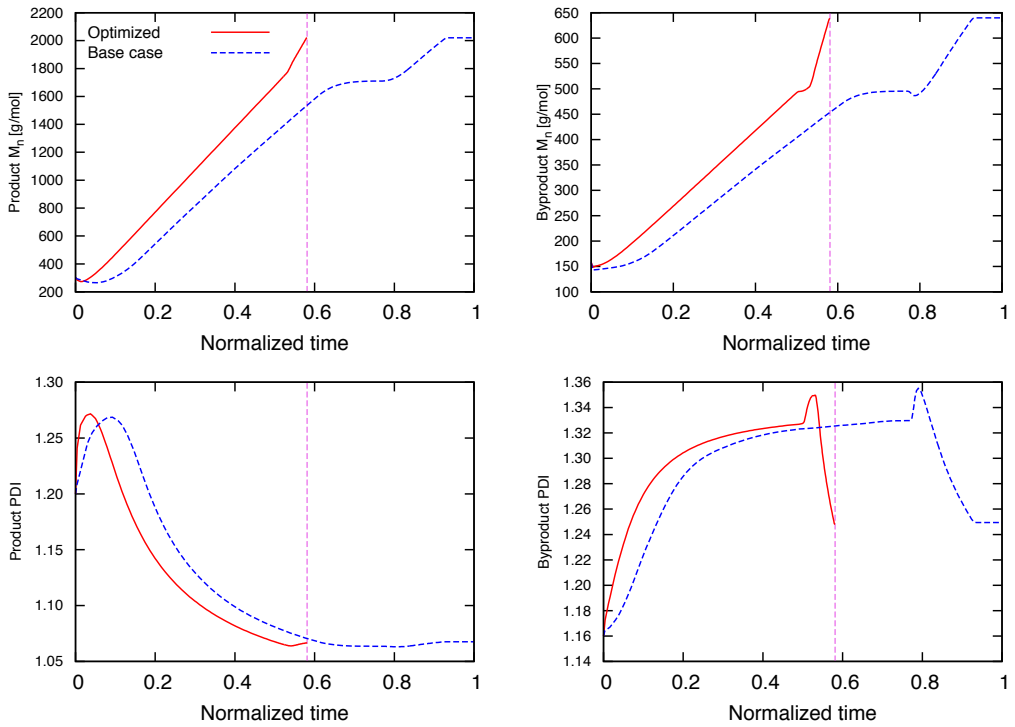



Figure 5.5: Polymer properties from the moment model: molecular weights and polydispersity indices

To demonstrate the product properties, we first show the number average molecular weight and PDI as functions of polymerization time in Fig. 5.5. The product polymers produced by the optimized recipe give the same number average molecular weight and a PDI very close to the base case products, which means the MWDs of the two cases are rather similar. For the byproduct, the differences between the two recipes are also minor. In addition, Fig. 5.6 shows the evolution of the two quality indices: impurity level and B ended chain ratio. The product from the optimized recipe shares the same quality level as the base case product. It is worth noting that the constraint on the impurity level is active for the optimized recipe, and relaxing the limit value may further reduce the batch time.

### Optimization with Population Balance Models

Recipe optimization over the PBM is a challenging task that requires significant computation effort. To facilitate the optimization, we adopt the optimized recipe from the

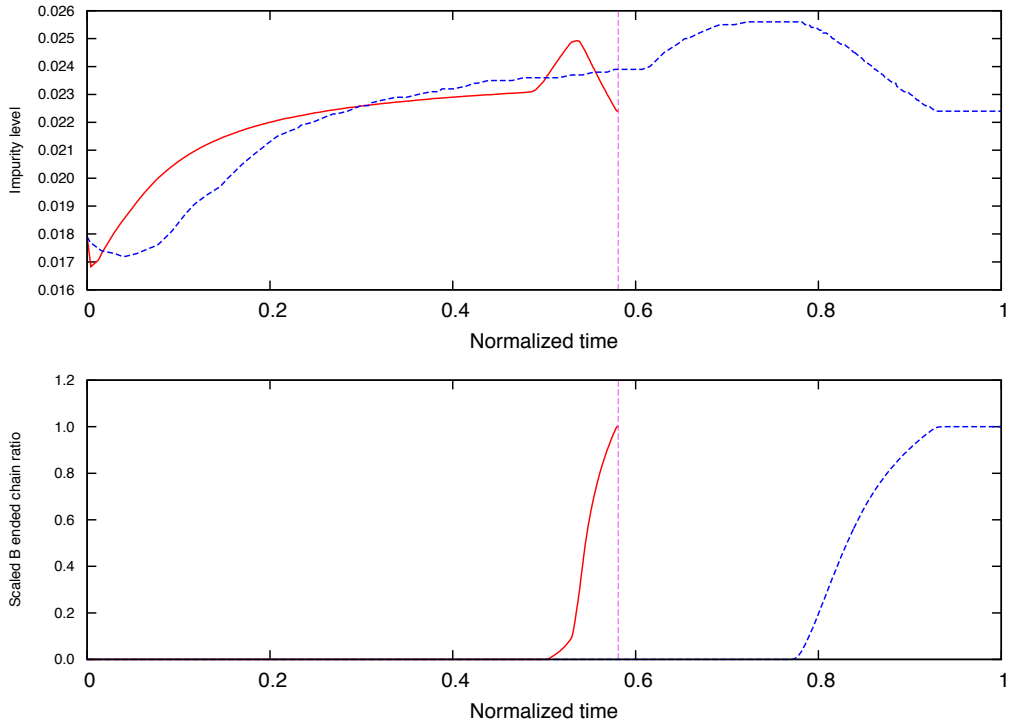


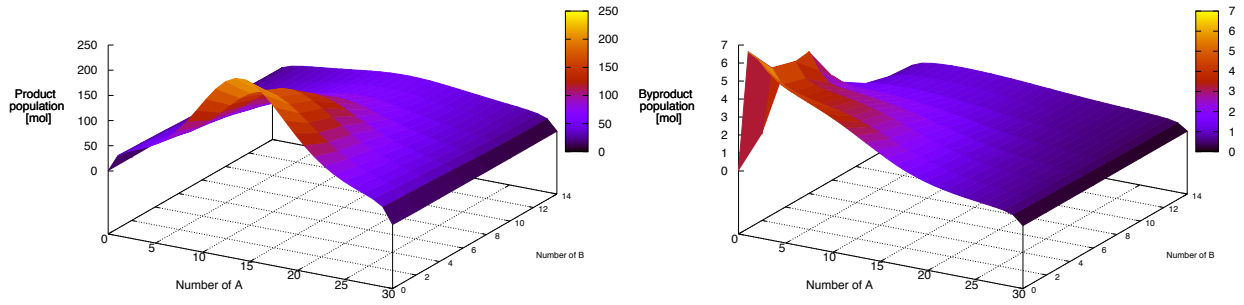

Figure 5.6: Polymer properties from the moment model: important quality indices

MM and start the optimizer at that solution. The initial guesses for population states are obtained through dynamic simulation beforehand, and it can be also verified that the optimal recipe generated by the MM remains a good feasible solution for the PBM. By this means, a large-scale optimization problem with the PBM is able to be solved by IPOPT, and the detailed information is listed in Tab. 5.3. The optimal batch time is slightly worse than the one obtained with the MM, and the end-point values of the final time process constraints are similar to the previous case. However, there is a huge expansion of the model scale in terms of variable and constraint numbers, in spite of a less dense discretization mesh being used. Consequently, the optimization problem becomes slow to solve. It should be noted that the base case recipe and the set of process constraints are chosen to illustrate the use of dynamic optimization and do not necessarily reflect the true capability or limitations of the plant. However, the actual potential reaction time saving can still be significant.

The optimal control recipe generated by using the PBM is very similar to the optimal

| Opt. soln (h) | # of var. | # of con. | CPU (s) |
|---|---|---|---|
| 0.588 | 191,408 | 185,147 | 5221 |

Table 5.3: Recipe optimization statistics and results from the population balance model




(a) Product chains

(b) Byproduct chains

Figure 5.7: Chain length distributions of the optimized recipe of the PBM

MM solution. In addition, the PBM model is able to reveal the chain length distribution with three-dimensional plots shown in Fig. 5.7 for the product and byproduct chains. The complete distribution information is helpful in predicting physical properties such as viscosity of the polymer product.

## 5.5 Concluding Remarks

We have developed a modeling and optimization methodology for semi-batch ring-opening polymerization processes for block copolymers, following two different approaches to model the polymer species: the population balance approach and method of moments. In the case study example, we first demonstrated that both reactor models were able to match real plant data adequately. The following recipe optimization results showed the potential of our modeling and optimization framework to enhance the process performance by redesigning the reactor operating policy. In particular, the moment model excelled with respect to computational performance. The population balance model can still be appreciated for its ability to uncover the chain length distribution.

The phase of reactor model development and recipe optimization is complete. The moment model is able to capture the dynamic characteristics of the polymerization operations with moderate computational complexity. Therefore, it is well suited for the integrated scheduling and dynamic optimization problem.

# Chapter 6

# Scheduling Method Development

In this chapter, we develop a scheduling optimization framework oriented for industrial applications. The problem scope is extended to a complete production process that has multiple units, manufacturing a variety of products. The scheduling method is based on the discrete time resource task network (RTN) representation. A number of modifications have been made to the conventional RTN models such as multi-extent resource balances, resource limit balances and resource slacks. The extended RTN model is further reformulated to the state space form by incorporating *lifted* state variables that represent task histories. The state space RTN model facilitates reactive schedule design, particularly when used with the rolling horizon scheme. The method can be applied to processes with mixed equipment types (continuous and batch) and general structures. The scheduling method, combined with the dynamic reactor models developed in earlier chapters, opens the gate to the integrated scheduling and dynamic optimization problem.

## 6.1 Background Information

The Dow Chemical Company has successfully applied discrete time scheduling models in optimizing many of its production facilities, especially with the resource task network (RTN) approach [154, 155, 156]. The discrete time RTN model has a concise structure of a few basic equations, but it is versatile enough to accommodate a wide array of different scheduling constraints without excessive customization. It also renders a low integrability gap in the resulting MILP that compensates the large number of binaries used [157]. Recently, a number of reformulation and solution algorithms [158, 159] have been investigated for discrete time scheduling formulations, which can dramatically improve their computational performance for large-scale problems.

In practice, optimized production schedules need to be updated continually in response to changes in product demands as well as process disruptions. Rescheduling capability is therefore required to achieve reliable scheduling implementations and minimize process performance losses in the presence of uncertainties and disruptions [160, 161]. This rescheduling procedure often starts with partitioning the set of tasks into smaller groups distinguished by attributes such as completeness and flexibility, and then re-optimizing by *partially modifying* the current schedule. In practice, rescheduling is preferred to be performed frequently in a periodic manner to keep the schedule *up-to-date*, which shares some typical characteristics of model predictive control (MPC) strategies, particularly the receding horizon scheme, where the optimization algorithm is constructed and solved repeatedly within a forwarding scheduling horizon. Subramanian et al. [162] showed a general procedure to translate well-known discrete time scheduling formulations to state space models that are the foundation for MPC algorithms, and they highlighted the advantage of the obtained state space models in modeling typical process disruptions.

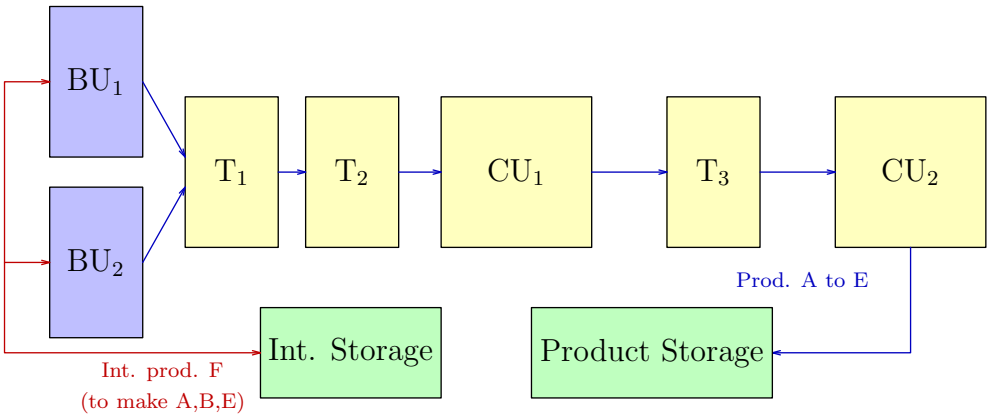



Figure 6.1: Process flowsheet of the mixed plant

## 6.2 Problem Description

We consider a representative example of mixed batch/continuous processes in the chemical industry, depicted in Fig. 6.1. As an example, the process consists of two batch processing units ($BU_1$, $BU_2$), three buffer storage tanks ($T_1$, $T_2$, $T_3$), and two continuous processing units ($CU_1$, $CU_2$). The two batch units in parallel have identical processing capabilities, and the buffer tanks and continuous units are in series. Five different products (Prod. A to E) are first processed by one of the batch units and then flow through the continuous buffer tanks and processing units into product storage. An intermediate product F is made in one batch unit and directly transferred to dedicated storage, and F is later used as a raw material to make Prod. A, B, and E. The process is operated under the following operating policies:

- Batch units are operated with fixed recipes (fixed task length and batch size).
- Buffer tanks have variable inlet/outlet flow rates.
- $CU_1$ and $CU_2$ have variable processing rates but no inventory.
- Mixing of products is not allowed in any units.
- Product transition time losses need to be considered for $CU_1$ and $CU_2$.
- Processing capacity of $CU_2$ is consumable and needs to be replenished *off-line*.

Challenges arise when considering modeling the mixed plant within the conventional discrete time RTN framework. On the one hand, the number of tasks is increased. For example, buffer tank $\mathrm{T}_1$ has two inlet and one outlet streams, and their flow rates are mutually independent. This requires three separate tasks to be defined instead of a single processing task. It also applies to the other tanks with inventories. In addition, the RTN model requires additional storage tasks to be defined to enforce operating rules such as non-mixing. Also, the replenishment task of $\mathrm{CU}_2$ involves two steps, where the unused capacity is first discarded and then the (resource level) of the available capacity is reset to full (e.g., changing a filter). Conventional RTN therefore requires two tasks to model the operation. Meanwhile, the number of integer assignment variables for a task is large, due to small time slots used to adjust processing rates frequently enough. Thus, reducing the number of tasks becomes an important issue especially for problems with long scheduling horizon. We apply the extended RTN models to accomplish the goal, where new continuous variables and balance equations are introduced such as task extents and resource limit balances. The required number of tasks in the extended RTN can be much fewer than the conventional one, e.g., single processing task for the tanks, no storage tasks for non-mixing, and one task for $\mathrm{CU}_2$ replenishment. On the other hand, the conventional RTN model is generally unsuitable for reactive scheduling as it does not track the history of tasks nor incorporate disturbance terms. We employ the state space form RTN model to overcome the limitation. The status of the process is captured by the state variables over the entire scheduling horizon. The scheduling algorithm can be restarted conveniently with updated state measures. Disturbances such as task delays can be explicitly modeled in the state space RTN model by defining corresponding model parameters. Reactive scheduling with the state space RTN model is able to optimally respond to changes in the production environment, maintaining the feasibility and optimality of production schedules through periodic updates.

## 6.3 RTN-based Scheduling Model

In discrete time RTN models, the known scheduling horizon is evenly divided to $H$ time slots with length $\Delta t$. The duration of a task is approximated by a rounded-up integer multiple of $\Delta t$, denoted by $\tau$. In this section, we first review the basic equations used in conventional RTN models [64, 65], followed by discussions on the RTN extensions and transformation to state space models.

### 6.3.1 Conventional RTN Models

The fundamental equation of the RTN model is the balance equation on the amount of *excess resources*, which are the resources not used by any currently active tasks at a given time:

$$R_{r,t} = R_{r,t-1} + \sum_{i\in\mathcal{I}}\sum_{\theta=0}^{\tau_i} \mu_{i,r,\theta} N_{i,t-\theta} + \sum_{i\in\mathcal{I}}\sum_{\theta=0}^{\tau_i} \nu_{i,r,\theta}\xi_{i,t-\theta} + \Pi_{r,t}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}. \tag{6.1}$$

Here, $R_{r,t}$ is the excess resource level of resource $r$ at time $t$, which depends on its previous level $R_{r,t-1}$, and interactions with tasks (two summation terms) and external influences $\Pi_{r,t}$. In the first interaction term, $N_{i,t}$ is an integer decision variable which represents the number of occurrence of task $i$ at time $t$, and $\mu_{i,r,\theta}$ denotes the integer interaction matrix between tasks and resources, where the time offset index $\theta$ ranges from $0$ to the task length $\tau_i$. The second interaction term represents the continuous task-resource interactions with coefficient $\nu_{i,r,\theta}$ and variable size $\xi_{i,t}$. The last participating term $\Pi_{r,t}$ can be used to model external supplies or product orders transferred into or out of the system. The excess resource levels in the resource balance equation (6.1) are constrained by resource limits:

$$R_{r,t}^{min} \leqslant R_{r,t} \leqslant R_{r,t}^{max}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}. \tag{6.2}$$

Finally, the batch sizes of active tasks are limited by equipment sizes, safety limits, etc., and this can be specified using the upper and lower limits of task sizes $V_i^{max}$ and $V_i^{min}$:

$$V_i^{min} N_{i,t} \leqslant \xi_{i,t} \leqslant V_i^{max} N_{i,t}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}. \tag{6.3}$$

Eqs. (6.1)-(6.3) comprise the core equations of a discrete time RTN model. An objective function can be added to formulate an MILP for scheduling optimization. Many economic objectives can be incorporated such as process profit maximization or cost minimization. Other indices such as makespan can be more complex to formulate, since the discrete time formulation requires a scheduling horizon with known length [163].

### 6.3.2 RTN Extensions

Wassick and Ferrio [155] have proposed a number of extensions to the conventional RTN model to improve its modeling capability and solution efficiency in handling industrial scheduling problems. Here we summarize a few extensions used in this study.

First, the external resource transfer variable $\Pi_{r,t}$ in the resource balance (6.1) can be used to model outgoing deliveries to fulfill a product order $o$ with known time window specified by its early date $E_o$ and due date $D_o$. If there are multiple orders requiring different products and the mapping is described by set $\Omega_{o,r}$, the order fulfillment constraint can be stated as follows:

$$\sum_{t \geq E_o}^{t \leqslant D_o} (-\Pi_{r,t}) = Q_{o,r}, \quad \forall (o,r) \in \Omega_{o,r}, \Pi_{r,t} \leqslant 0, \tag{6.4}$$

where $Q_{o,r}$ is the product resource demand.

In a RTN model, a task can be defined to have multiple steps and/or resource interaction routes. In these cases, multiple batch sizing variables need to be introduced for a single task to determine the batch extents, where an extent refers to the processed material

amount in a task step or resource interaction route. The resource balance equation can be modified as

$$R_{r,t} = R_{r,t-1} + \sum_{i\in\mathcal{I}}\sum_{\theta=0}^{\tau_i} \mu_{i,r,\theta} N_{i,t-\theta} + \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}}\sum_{\theta=0}^{\tau_i} \nu_{i,n,r,\theta}\xi_{i,n,t-\theta} + \Pi_{r,t}, \quad \forall r\in\mathcal{R}, t\in\mathcal{T}. \tag{6.5}$$

An additional summation over task extent index $n$ is added for the continuous task-resource interaction term, with the disaggregated interaction parameter $\nu_{i,n,r,\theta}$ and variable $\xi_{i,n,t}$. Consequently the batch size limit constraint (6.3) is rewritten with respect to each extent:

$$V_{i,n}^{min} N_{i,t} \leqslant \xi_{i,n,t} \leqslant V_{i,n}^{max} N_{i,t}, \quad \forall i\in\mathcal{I}, n\in\mathcal{N}, t\in\mathcal{T}. \tag{6.6}$$

The multi-extent concept may avoid defining separate tasks for a group of operations that take place jointly. Later we show the use of multi-extents in the RTN modeling of continuous units.

In a conventional RTN model, a task can only influence excess resource levels while the resource limits $R_{r,t}^{min}$ and $R_{r,t}^{max}$ are given constant model parameters. However, it is advantageous for tasks to be able to interact with resource limits for material transfer and storage operations as the resource storage limits change over time. For this purpose, two resource limit balance equations can be constructed in analogy to the resource balance equation (6.5):

$$R_{r,t}^{max} = R_{r,t-1}^{max} + \sum_{i\in\mathcal{I}}\sum_{\theta=0}^{\tau_i} \alpha_{i,r,\theta} N_{i,t-\theta} + \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}}\sum_{\theta=0}^{\tau_i} \beta_{i,n,r,\theta}\xi_{i,n,t-\theta}, \quad \forall r\in\mathcal{R}, t\in\mathcal{T}; \tag{6.7a}$$

$$R_{r,t}^{min} = R_{r,t-1}^{min} + \sum_{i\in\mathcal{I}}\sum_{\theta=0}^{\tau_i} \alpha'_{i,r,\theta} N_{i,t-\theta} + \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}}\sum_{\theta=0}^{\tau_i} \beta'_{i,n,r,\theta}\xi_{i,n,t-\theta}, \quad \forall r\in\mathcal{R}, t\in\mathcal{T}. \tag{6.7b}$$

The integer and continuous interaction parameters $\alpha_{i,r,\theta}$ ($\alpha'_{i,r,\theta}$) and $\beta_{i,n,r,\theta}$ ($\beta'_{i,n,r,\theta}$) are introduced, similar to those defined for changing resource levels ($\mu_{i,r,\theta}$ and $\nu_{i,n,r,\theta}$ in the resource balance (6.5)).

The excess resource limit constraint (6.2) can bring difficulties in solving RTN models in some highly resource-constrained scenarios, and the MILP algorithm only returns a null solution indicating problem infeasibility. A modification can be made on Eq. (6.2) to relax the bounds on the resource limits:

$$R_{r,t}^{min} - S_{r,t}^{min} \leqslant R_{r,t} \leqslant R_{r,t}^{max} + S_{r,t}^{max}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}. \tag{6.8}$$

By adding positive slack variables $S_{r,t}^{min}$ and $S_{r,t}^{max}$, the resource level is allowed to exceed the presumed bounds, but the violation should be penalized in the scheduling objective. Extended RTN models with the *soft* resource limit constraint (6.8) can provide more informative solutions for analyzing constraining resources of the process.

### 6.3.3 State Space RTN Model

State space models are widely used in process control literature. A state space model represents a dynamic process system by a set of first-order differential/difference equations including three groups of variables, the input, output, and state variables. The input variables are the decision variables and the output variables are associated with control goals. The state of the dynamic system at any given time can be fully represented by state variables, and the future state can be forecast by the current state and input variables of the system. The key state evolution equations for the extended RTN model are Eqs. (6.5) and (6.7), where the input variables are task assignment and batch size decisions, and the state variables include resource levels and associated limits. However, these states can be influenced by input variables at earlier steps as shown in Eqs. (6.5) and (6.7) (when task length $\tau_i > 1$), which essentially represent high-order dynamics. It follows that the current set of state variables is not sufficient to represent the entire system state. To solve this problem, the state vector needs to be augmented through *lifting*, i.e., introducing new

state variables [162]:

$$\bar{N}_{i,t,\theta} \triangleq N_{i,t-\theta}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, 0 \leqslant \theta \leqslant \tau_i; \tag{6.9a}$$

$$\bar{\xi}_{i,n,t,\theta} \triangleq \xi_{i,n,t-\theta}, \quad \forall i \in \mathcal{I}, n \in \mathcal{N}, t \in \mathcal{T}, 0 \leqslant \theta \leqslant \tau_i. \tag{6.9b}$$

$\bar{N}_{i,t,\theta}$ and $\bar{\xi}_{i,n,t,\theta}$ record the decision history of the system. For instance, $\bar{N}_{i,t,\theta} = 1$ means that there is one instance of task $i$ that has been running for $\theta$ time periods at time $t$. Or equivalently, it can be interpreted as an active task $i$ at time $t$ was started $\theta$ slots ago. Note that in definition (6.9) the augmenting states are equivalent to the input variables $N_{i,t}$ and $\xi_{i,n,t}$ when $\theta = 0$. The sufficient input and state vectors can be identified as:

Inputs: task assignment $N_{i,t}$ and task (extent) size $\xi_{i,n,t}$

States: resource level $R_{r,t}$, limit levels $R^{max}_{r,t}$ and $R^{min}_{r,t}$, and task history $\bar{N}_{i,t,\theta}$ and $\bar{\xi}_{i,n,t,\theta}$

The set of first-order dynamic state evolution equations can be stated as follows:

$$\bar{N}_{i,t,\theta} = \bar{N}_{i,t-1,\theta-1}, \tag{6.10a}$$

$$\bar{\xi}_{i,n,t,\theta} = \bar{\xi}_{i,n,t-1,\theta-1}, \tag{6.10b}$$

$$R_{r,t} = R_{r,t-1} + \sum_{i\in\mathcal{I}}\sum_{\theta=0}^{\tau_i} \mu_{i,r,\theta}\bar{N}_{i,t,\theta} + \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}}\sum_{\theta=0}^{\tau_i} \nu_{i,n,r,\theta}\bar{\xi}_{i,n,t,\theta} + \Pi_{r,t}, \tag{6.10c}$$

$$R^{max}_{r,t} = R^{max}_{r,t-1} + \sum_{i\in\mathcal{I}}\sum_{\theta=0}^{\tau_i} \alpha_{i,r,\theta}\bar{N}_{i,t,\theta} + \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}}\sum_{\theta=0}^{\tau_i} \beta_{i,n,r,\theta}\bar{\xi}_{i,n,t,\theta}, \tag{6.10d}$$

$$R^{min}_{r,t} = R^{min}_{r,t-1} + \sum_{i\in\mathcal{I}}\sum_{\theta=0}^{\tau_i} \alpha'_{i,r,\theta}\bar{N}_{i,t,\theta} + \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}}\sum_{\theta=0}^{\tau_i} \beta'_{i,n,r,\theta}\bar{\xi}_{i,n,t,\theta}. \tag{6.10e}$$

Eqs. (6.10a) and (6.10b) are derived from the definitions of task history states in Eqs. (6.9). The external resource transfer term $\Pi_{r,t}$ in Eq. (6.10c) can be treated as a disturbance term to the system. The state space RTN model can be constructed after replacing Eqs. (6.5) and (6.7) with Eqs. (6.10). The resulting state space model is inherently favorable for recursive scheduling with the receding horizon scheme. Performing a rescheduling task

only requires knowing the initial values of all the state variables, which can be readily obtained from the previous scheduling optimization solution. This can be recognized as a systematic and automatic way of carrying out task set partitioning.

Moreover, the state space RTN mode can efficiently represent process disturbances through modifying the state evolution equations. Examples on a number of disturbance types were studied by Subramanian et al. [162]. Here we focus on the most common disruption source, task delays, but with a different modeling approach than that used by Subramanian et al. [162]. We use a simple example to illustrate the idea. Let an arbitrary task $i$ of length $\tau_i = 2$ start at time $t = 0$. If no delays occur the evolution of the discrete task history state $\bar{N}_{i,t,\theta}$ is shown in Fig. 6.2(a), where the bold face ones are active. In contrast, if the task is delayed for one time period from $t = 1$ to $t = 2$, the states at $t = 2$ should be updated differently as shown in Fig. 6.2(b). Mathematically, the task delay event can be described by a series of fixed binary disturbance parameters $d_{i,t,\theta}$. $d_{i,t,\theta} = 1$ indicates a unit time period delay of task $i$ at time $t$ (and task $i$ starts at time $t - \theta$). For instance, $d_{i,1,1} = 1$ and others are zero in the simple example. The state space equation for the discrete task history state (6.10a) can be modified to include task delays:

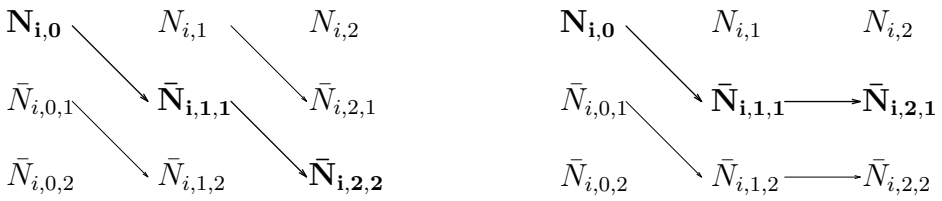
$$\bar{N}_{i,t,\theta} = (1 - \sum_{\theta' \geqslant 0}^{\theta' \leqslant \theta} d_{i,t-1,\theta'})\bar{N}_{i,t-1,\theta-1} + \sum_{\theta' \geqslant 0}^{\theta' \leqslant \theta} d_{i,t-1,\theta'}\bar{N}_{i,t-1,\theta}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, 1 \leqslant \theta \leqslant \tau_i. \quad (6.11)$$

The evolution equations for the continuous history state can be rewritten in a consistent way:

$$\bar{\xi}_{i,n,t,\theta} = (1 - \sum_{\theta' \geqslant 0}^{\theta' \leqslant \theta} d_{i,t-1,\theta'})\bar{\xi}_{i,n,t-1,\theta-1} + \sum_{\theta' \geqslant 0}^{\theta' \leqslant \theta} d_{i,t-1,\theta'}\bar{\xi}_{i,n,t-1,\theta}, \quad \forall i \in \mathcal{I}, n \in \mathcal{N}, t \in \mathcal{T}, 1 \leqslant \theta \leqslant \tau_i. \quad (6.12)$$

Longer task delays that take multiple time periods ($\tau_d > 1$) can be modeled by consecutive disturbance parameters $d_{i,t,\theta}, d_{i,t+1,\theta} \ldots d_{i,t+\tau_d-1,\theta}$.

The state space RTN model with extensions we have developed consists of




(a) No task delay (b) Task delay at $t = 1$

Figure 6.2: Effect of task delays on the evolution of task history states

State dynamics: Eqs. (6.10c), (6.10d), (6.10e), (6.11), (6.12).

Process constraints: Eqs. (6.4), (6.6), (6.8).

These equations essentially model the linear dynamics of the scheduling system that serve as the core constraints in scheduling optimization formulations. The form of the scheduling objective can be different from one application to another, and we show the objective function for the mixed process after explaining the process RTN representation.

## 6.4 Process RTN Representation

The RTN representation of the mixed batch/continuous process shown in Fig. 6.1 can be developed by translating the associated operations to bipartite graphs of task and resource nodes. We illustrate this in parts: the batch processing units, buffer tanks, and continuous processing units. No final product storage tasks are explicitly considered, because product inventory levels and capacities are modeled as excess resource levels and corresponding limits. We use the RTN representations associated with processing Prod. A to demonstrate the representation approach, because RTN representations of other products are very similar.

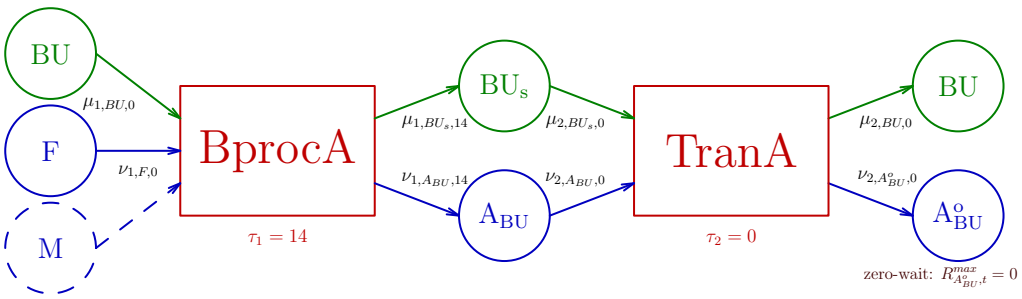



Figure 6.3: RTN representation of the batch processing unit

### 6.4.1 Batch Processing Units

The batch units are the first process stage for all products, and they can also be used as temporary storage facilities when a batch is made but the downsteam buffer tank is not available (at its capacity or occupied by other products). We show the RTN representation of one of the batch units in Fig. 6.3. Two tasks are defined, corresponding to the major batch operation (BprocA) and the material transfer operation (TranA). Task BprocA lasts for $14$ time intervals ($\tau_1 = 14$), consumes the batch unit $\mathrm{BU}$, intermediate Int. F, and other materials $\mathrm{M}$ at the beginning, and releases a filled unit $\mathrm{BU_S}$ loaded with raw product $\mathrm{A_{BU}}$ at the end. The amount of other materials $\mathrm{M}$ is not accounted in resource balances as we assume $\mathrm{M}$ is unlimited. Task TranA is assumed to be instantaneous ($\tau_2 = 0$) that corresponds to transferring the temporarily stored resource $\mathrm{A_{BU}}$ to the downstream buffer tank, and the unit is therefore *cleaned* and ready to start the next batch. This translates to converting resources $\mathrm{BU_S}$ and $\mathrm{A_{BU}}$ to $\mathrm{BU}$ and $\mathrm{A^o_{BU}}$ in the RTN language. The values of the resource task interaction parameters are listed in Tab. D.2 in the Appendix C. It is worth noting that a zero-wait policy is enforced on $\mathrm{A^o_{BU}}$ by setting its upper limit equal to zero. This guarantees the transfer task only takes place when the buffer tank is available to store $\mathrm{A^o_{BU}}$.

The RTN graph applies to both batch units since they are identical and representations for making other products (B to E) are similar to Fig. 6.3, with slight differences in the material resources consumed and generated. The transfer task is not introduced for Int. F

since it does not go through the downstream processing stages but is stored in dedicated tanks after the batch processing step and recycles back to the batch units as the initial material for making other products. Fixed batch sizes of the processing and transfer tasks are assumed, and the sizes of the two tasks are equal to ensure the transfer task empties the batch unit. An alternative general approach to model temporary storage of batch products is reported by Castro et al. [164], where hold-in-storage tasks are introduced in contrast to the transfer task.

### 6.4.2 Buffer Tanks

The buffer tanks are simple continuous units based on mass balance, and therefore the material transfers between its inlet flow, inventory, and outlet flow need to be described by the RTN. Taking the first tank $\mathrm{T}_1$ as the example, the RTN representation is depicted in Fig. 6.4. The buffering task spans a single time slot and seizes the equipment resource $\mathrm{T}_1$ during execution. In order to model mass transfers between the two inlets $\mathrm{A}^{\mathrm{o}}_{\mathrm{BU}_1}$ and $\mathrm{A}^{\mathrm{o}}_{\mathrm{BU}_2}$, inventory $\mathrm{A}_{\mathrm{T}_1}$, and outlet $\mathrm{A}^{\mathrm{o}}_{\mathrm{T}_1}$, the multi-extent concept is applied. Three extents are defined where Extent 1 and 2 model the streams from the batch units into the tank, and Extent 3 represents the outgoing flow to downstream. Note that the interaction between the task and material resources are assumed to be instantaneous. More specifically, the continuous task-resource interaction parameters $\nu_{i,n,t,\theta}$ have no time offsets ($\theta = 0$) on both the consumption and the generation sides. This means the material balances are taken into account at the beginning of the buffering task even if the transfer process continues steadily within the time slot. The continuous flow rate can be back-calculated with the extent size and time grid length if a predefined flow rate profile (piece-wise constant for example) is assumed. The RTN representations for buffering other products are of similar structures. In addition, the resource limit balance is used to ensure the buffer tank can store at most one type of product at a time. In this example, the excess resource capac-

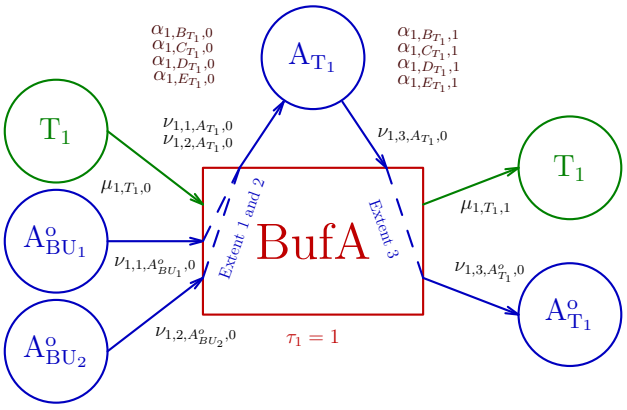



Figure 6.4: RTN representation of buffer tank $\mathrm{T}_1$

ity of products other than Prod. A in the buffer tank are temporarily removed when task BufA starts, and restored when it finishes. This is carried out by appropriately defining $\alpha_{i,r,\theta}$ in the maximum limit balance (6.10d) for the inventory resources of other products (analogous to $\mathrm{A}_{\mathrm{T}_1}$, e.g., $\mathrm{B}_{\mathrm{T}_1}$, see Fig. 6.4). The other buffer tank $\mathrm{T}_2$ can be modeled with a similar task with two extents, which eventually converts $\mathrm{A}^{\mathrm{o}}_{\mathrm{T}_1}$ to $\mathrm{A}^{\mathrm{o}}_{\mathrm{T}_2}$.

### 6.4.3 Continuous Processing Units

We show the RTN representation of the two continuous processing units $\mathrm{CU}_1$ and $\mathrm{CU}_2$ together with the buffer tank $\mathrm{T}_3$ in Fig. 6.5. The first continuous processing unit $\mathrm{CU}_1$ processes the material flow at desired rates by task $\mathrm{C}_1\mathrm{procA}$, and the material interactions are also assumed to be instantaneous. Only one task extent is needed since $\mathrm{CU}_1$ allows no inventory. Next, the buffer tank $\mathrm{T}_3$ has very similar RTN structure to that of $\mathrm{T}_1$. For $\mathrm{CU}_2$, a material resource $\mathrm{CAP}$ is defined to represent its consumable processing capacity. Task $\mathrm{C}_2\mathrm{procA}$ consumes both $\mathrm{A}^{\mathrm{o}}_{\mathrm{T}_3}$ and CAP to produce the final Prod. A. When the remaining processing capacity becomes low, the replenishment task $\mathrm{C}_2\mathrm{repln}$ is performed. This task lasts for two time periods, and the two task extents involved take place at the beginning and end of the task, respectively. The first extent empties the remaining unused resource level of $\mathrm{CAP}$ at $\theta = 0$, which is achieved via forcing the upper bound of $\mathrm{CAP}$ to $0$ in

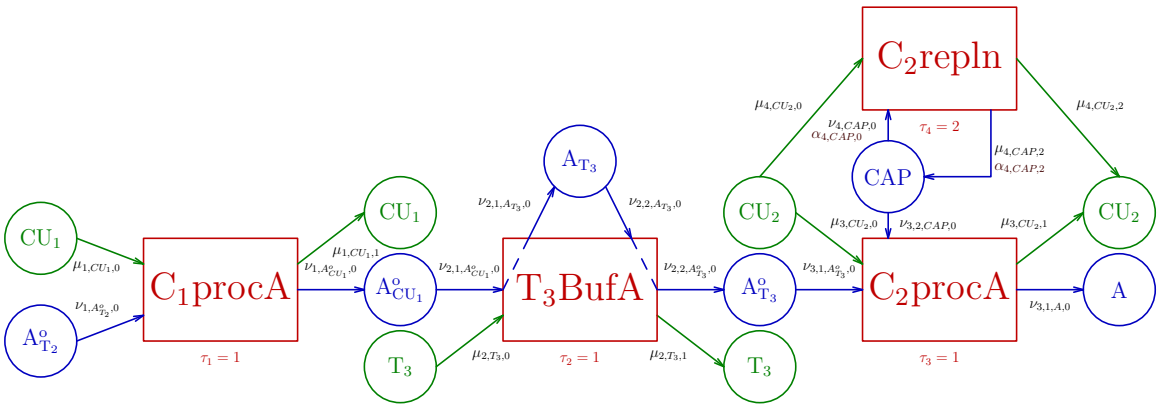



Figure 6.5: RTN representation of the continuous processing units

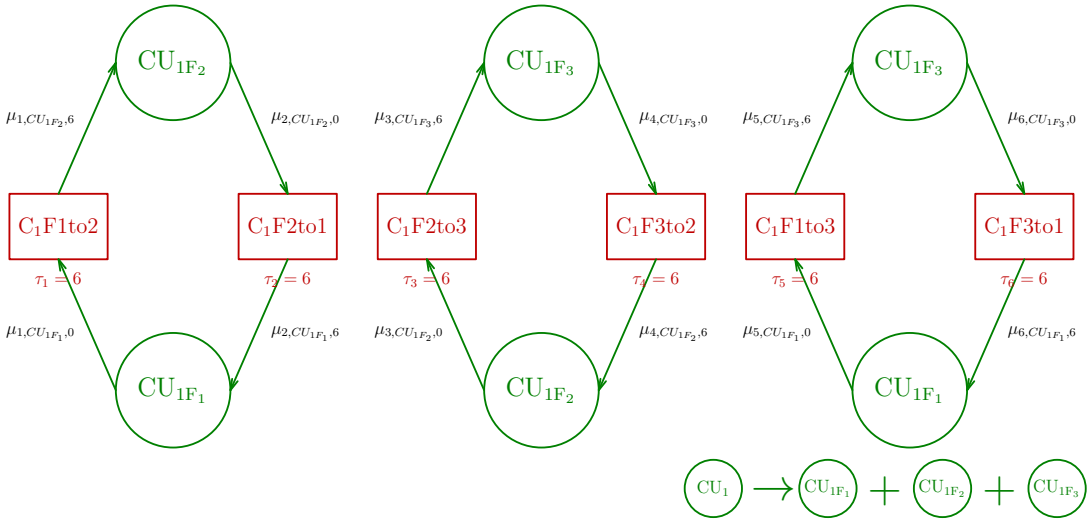



Figure 6.6: RTN representation of the changeover tasks for $\mathrm{CU}_1$

the resource limit balance. This allows for a variable amount of remaining $\mathrm{CAP}$ to be removed, and the threshold value for admissible replenishment actions can be set by specifying the batch extent limit. For example, the extent limit can be set to $25\%$ of the maximum resource level of $\mathrm{CAP}$, which renders degrees of flexibility for scheduling the replenishment task. The second extent restores the resource level of $\mathrm{CAP}$ to its maximum along with the resource level upper bound when task ends at $\theta = 2$. This task also seizes $\mathrm{CU}_2$ to ensure the unit is offline during capacity restoration.

However, the continuous processing units $\mathrm{CU}_1$ and $\mathrm{CU}_2$ have discrete behaviors when transition time losses are considered. For this mixed process, three different product fam-

ilies require equipment cleaning for changeovers. A way to address this modeling challenge is to introduce changeover tasks along with associated *dummy equipment resources*. For the example of the first continuous processing unit, the original equipment resource $\mathrm{CU}_1$ in Fig. 6.5 can be replaced with three dummy ones $\mathrm{CU}_{1\mathrm{F}_1}$, $\mathrm{CU}_{1\mathrm{F}_2}$, and $\mathrm{CU}_{1\mathrm{F}_3}$ for the three different product families, respectively. The processing task for a product belonging to a particular product family needs to consume and generate the corresponding dummy equipment resource. The unit is alternatively represented by the three dummy equipment resources, and the summation of their resource levels is no greater than one within the scheduling horizon. Then six changeover tasks are defined to toggle the unit between the three states for producing products in different categories, shown in Fig. 6.6. The lengths of the changeover tasks indicate the transition time loss matrix. Although in this case, the time losses are the same for all changeovers, the method is able to handle sequence-dependent changeovers with asymmetric transition matrices. Operation cost for performing changeover tasks can be explicitly considered in the scheduling objective function if necessary. The changeover tasks for $\mathrm{CU}_2$ transitions can be similarly defined, while the equipment resource $\mathrm{CU}_2$ is retained for scheduling the replenishment task.

## 6.5 Optimal Scheduling Formulation

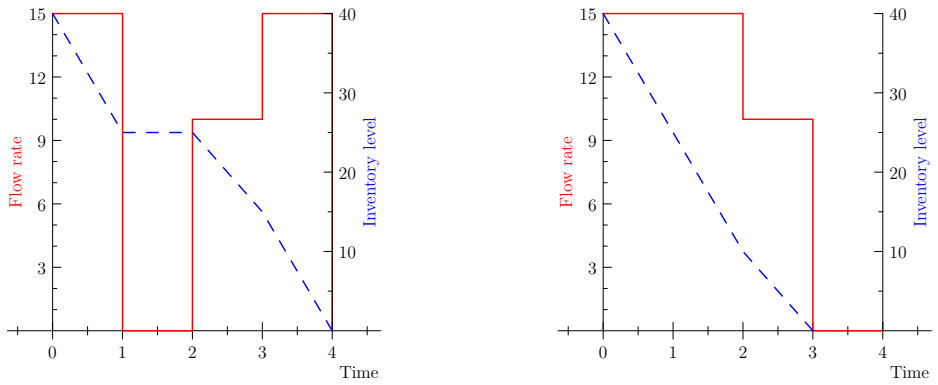
### 6.5.1 Scheduling Objectives

The optimal scheduling algorithm is expected to deliver optimized schedules best satisfying multiple manufacturing targets. The objective function varies with different scenarios. However, a number of basic elements are essential in most scheduling applications, such as order fulfillment and safety stock maintenance. If we assume an order must be delivered within a particular time window as Eq. (6.4) states, an economic value function can be associated with the filling amount term $-\Pi_{r,t}$, and the value function can be

time dependent to encourage early order retrievals. For managing product inventories, the slack variables $S_{r,t}^{min}$ in Eq. (6.8) are used at $t = H$, which allows mild violations on the product safety stock. However, the final time product resource levels are desired to stay within/close to the unrelaxed feasible region. This is achieved by penalizing the slack variable corresponding to the last time interval in the objective. An objective function combining the scores on order filling and resource level might be sufficient for driving a conventional batch scheduling problem for profit maximization/cost minimization. However, as continuous units are involved, further considerations are required. For instance, the processing rate profiles of the continuous units can be oscillatory as all the continuous processing tasks are narrowed down to a unit length in time such that the rate decision in a time slot is isolated from its neighbors. A comprehensive objective function should also take such issues into account, and resolve them by penalizing problematic operation postures.

### 6.5.2 Treatment of Flow Rate Profiles

Oscillatory flow rate profiles can occur when continuous processing units are operated under capacity. For example, when tank $\mathrm{T}_2$ is required to empty its current inventory $\mathrm{A}_{\mathrm{T}_2}$ within $4$ time periods, a flow rate profile may occur as shown in Fig. 6.7(a) ($\mathrm{T}_2$ max. output capacity = $15 \times 4 = 60$ and current inventory = $40$). This profile is "optimal" from the scheduling perspective, since the material transfer objective is achieved in time. However, the flow rate control leads to unnecessary switches that are undesirable in practice. This oscillatory profile is non-optimal from the equipment control point of view and it also propagates to the first continuous unit $\mathrm{CU}_1$. In this case, a schedule can have multiple flow rate profile solutions even in a local part of the complete schedule (see Fig. 6.7(a)) because the problem is under-constrained. A possible way to overcome this problem is to carry out post-optimization amendments. However, it can be a cumbersome process for




(a) A possible "optimal" profile (b) Regularized optimal profile

Figure 6.7: Comparison of different "optimal" flow rate profiles

plant schedulers to identify gaps between one flow after another. An alternative approach is to regularize the flow by penalizing the inventory resource levels in the buffer tanks over time. The inventory penalties force the material flow to go through the continuous processing stages as fast as possible, which can lead to steadier flow rate shapes. Here, the area under the inventory curve is minimized by the regularized flow in Fig. 6.7(b) for the illustrative example. The penalty coefficients should be small to reduce their effect on production scheduling decisions.

### 6.5.3 Mathematical Formulation

A comprehensive objective function used for the mixed process scheduling problem can be constructed as follows:

$$\max \phi = \sum_{(o,r)\in\Omega_{o,r}} \sum_{t\geq E_o}^{t\leq D_o} c_{o,r,t}^{order}(-\Pi_{r,t}) - \sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}} c_i^{tr} N_{i,t} - \sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} c_r^{inv} R_{r,t} - \sum_{r\in\mathcal{R}} c_r^{sl} S_{r,H}^{min}. \quad (6.13)$$

All the economic coefficients are positive, where $c_{o,r,t}^{order}$ represents the order value, $c_i^{tr}$ is the economic cost of product transition tasks, $c_r^{inv}$ indicates the inventory resource penalty in the buffer tanks, and $c_r^{sl}$ represents the safety stock violation penalty for product re-

sources. We use the state space RTN model as the model constraints. The developed scheduling formulation is a standard MILP problem.

## 6.6 Case Study

For the mixed process in this study, a long scheduling prediction horizon is generally preferred to manage product sequences and transitions, while small discretized time intervals are needed to capture the flow rate variations of the continuous units. These easily lead to a large number of integer variables in the scheduling formulation, and therefore solving the model to global optimality becomes difficult. However, reasonably good feasible solutions (say less than five percent optimality gap) are often acceptable for practical scheduling problems. Moreover, solution speed becomes more important than optimality when rescheduling in response to unscheduled process disruptions. For this reason, we allow a five percent optimality gap for our case study problems, which keeps the solution time within ten CPU minutes or less. We illustrate the scheduling algorithm with the rolling horizon scheme, where a nominal schedule is first designed and then rescheduling is performed with the shifted time window dealing with revealed unexpected events.

### 6.6.1 Nominal Schedule Design

For the first part, an optimal three-day schedule with one-hour time slots is obtained via solving the scheduling formulation to meet the customer demand of Day 1-3. In addition, the plant starts with initial inventories of Prod. A in $\mathrm{T}_2$ and Prod. B in $\mathrm{T}_3$ ($R_{\mathrm{T_2A},0} = 70$ and $R_{\mathrm{T_3B},0} = 70$). The model and solution statistics are given in Tab. 6.1. Gurobi finds a good feasible schedule with $\sim 3\%$ optimality gap and times out after 600s. The process Gantt chart is shown in Fig. 6.8, where the task rectangles are colored with respect to products/events and their widths represent task lengths. For the batch units $\mathrm{BU}_1$ and $\mathrm{BU}_2$,

| Opt. obj. | Gap (%) | Var. (Discrete) # | Con. # | Solver | CPU(s) |
|---|---|---|---|---|---|
| 563.0 | 2.6 | 134,148(4,380) | 140,382 | Gurobi 5.5.0 | 600 |

Table 6.1: Model and solution statistics of the nominal case

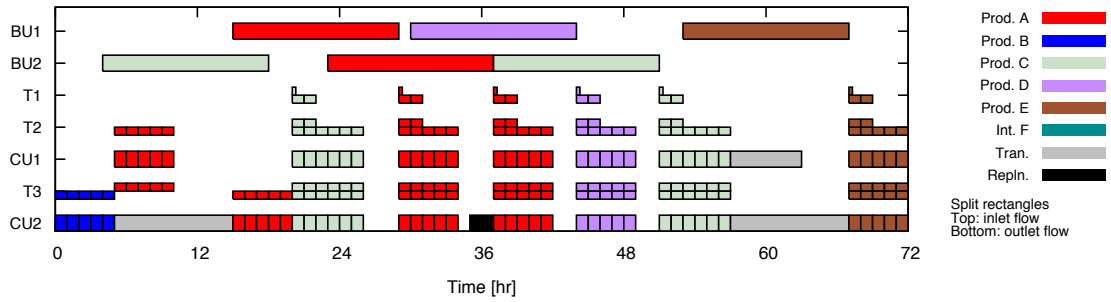
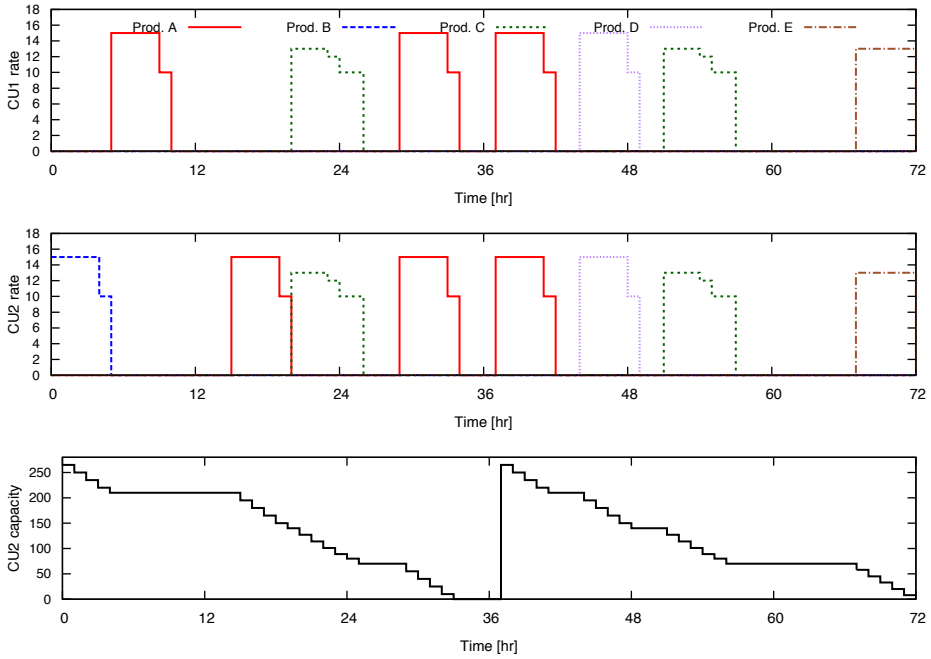


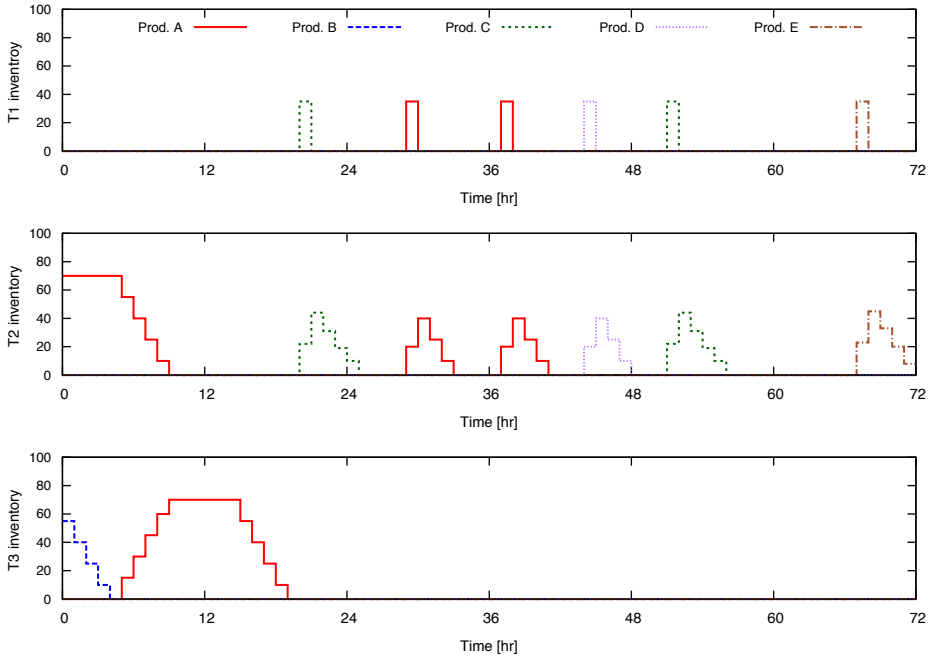

Figure 6.8: Optimal schedule for the mixed plant for Day 1-3

two batches of Prod. A and C, and one batch of Prod. D and E are scheduled. Split rectangles are employed to show the schedule of buffer tanks as their inlet and outlet flows are mutually independent in time. Note that short bars are used to record the appearance of $T_1$ inlet flows, whose material transfer time is assumed to be zero. Prod. transitions are observed in both schedules of $CU_1$ and $CU_2$ (gray rectangles), while $CU_2$ replenishment task is scheduled within the second day (black rectangles).

Besides the schedule, the model solution also includes operating details of the continuous units, such as the operating policies of $CU_1$ and $CU_2$ and the inventory profiles of $T_1$, $T_2$, and $T_3$, shown in Fig. 6.9(a) and Fig. 6.9(b), respectively. These profiles can be set as the reference trajectories for process monitoring and control purposes. Finally, Fig. 6.10 shows the evolution of the intermediate and final product (A to F) inventories. All of the final product inventory levels are well maintained around their initial levels at the end of the third day, and the level of the Int. F decreases as it is used to produce Prod. A, B, and E in the batch units.




(a) Continuous processing units




(b) Buffer tanks

Figure 6.9: Operating details of the continuous processing units and buffer tanks

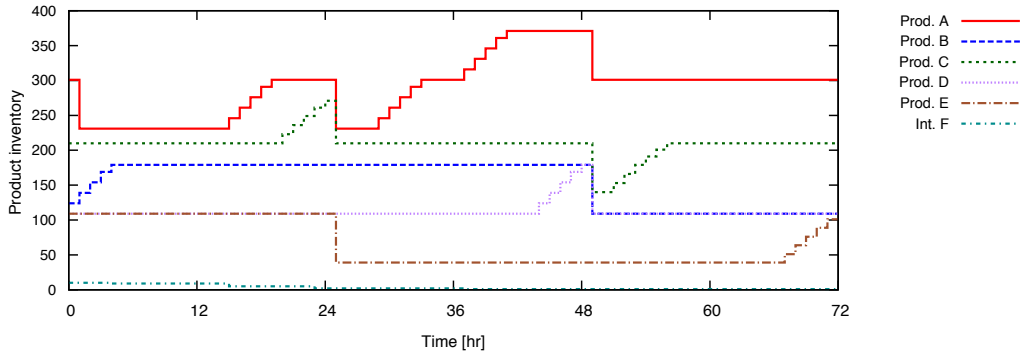



Figure 6.10: Product inventory profiles

### 6.6.2 Reactive Schedule Design

We illustrate the rescheduling capability within a rolling horizon scheme, where the scheduling time window is shifted forward by 24 hours. If the schedule in Fig. 6.8 has been executed without any upsets, the scheduling algorithm can be restarted by the end of Day 1, by solving a new iteration that incorporates the product demands of Day 2-4. The unfinished tasks by the end of Day 1 such as those in the batch units can be easily carried over to the new scheduling iteration within the initialization of the task history states. Three different scenarios are considered: 1). no process disruptions, 2). task delay in batch unit $\mathrm{BU}_1$, and 3). maintenance of buffer tank $\mathrm{T}_3$. The optimized schedules are shown in Figs. 6.11 in comparison and solution statistics are given in Tab. 6.2.

#### Scenario 1: no process disruptions

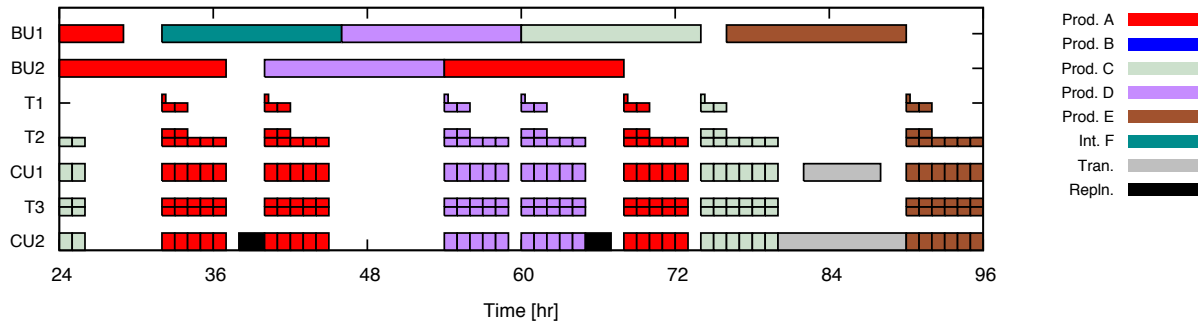
Fig. 6.11(a) gives the updated Gantt chart of the plant schedule from Day 2 to Day 4 with no disruptions. The scheduling algorithm has made a number of modifications to the previous schedule, such as adding batches of Prod. A and D as well as Int. F, due to the newly obtained knowledge of product demands of Day 4. Also, the transition from the previous schedule is smooth such that no drastic changes need to be made for the first couple of hours of Day 2.

**Scenario 2: task delay in $\mathrm{BU}_1$**

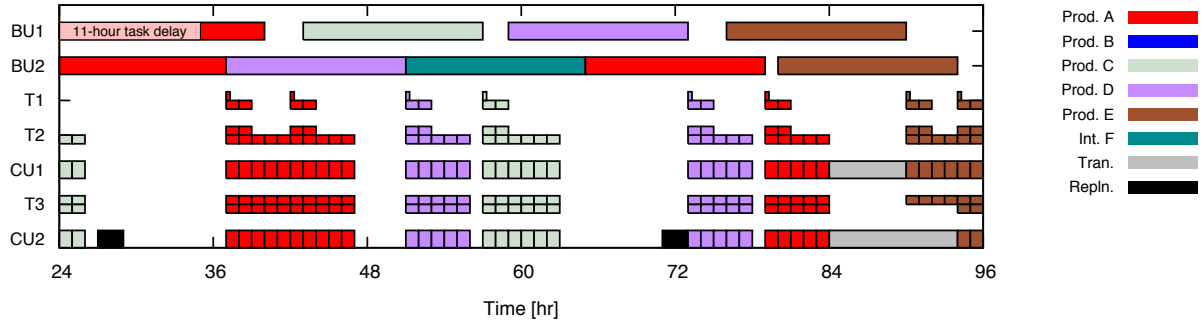
More interestingly, Fig. 6.11(b) shows the optimal schedule designed given the notice of task delay in $\mathrm{BU}_1$ for 11 hours. The optimal schedule now assigns the task of producing Int. F to $\mathrm{BU}_2$ in working order. Note that an additional batch is scheduled in Day 4 to produce Prod. E that is on back order. However, the demand for Prod. E is only one batch. The over-production is due to a modification made to safety stock constraint for the product resources. For example, the final time inventory of Prod. E used in the safety stock constraint accounts not only for the current actual stock level, but also the "raw products" that exist in $\mathrm{T}_1$, $\mathrm{T}_2$, and $\mathrm{T}_3$. The effective resource level can be calculated as a weighted summation of all Prod. E inventories, where the weights are noted as discount factors $f_r \in [0, 1]$. Here, $f_{\mathrm{E}} = 1$ for the final products, and $f_{\mathrm{T_{1E}}} < f_{\mathrm{T_{2E}}} < f_{\mathrm{T_{3E}}} < 1$ for the raw products. The effective resource level is used to encourage upstream tasks that are not available to deliver final products before the end of the current scheduling time horizon, but their corresponding downstream operations can be finished in the immediate future. Nevertheless, the over-production may later be adjusted at the next scheduling iteration within the rolling horizon scheme, depending on the newly revealed demand information.

**Scenario 3: maintenance of $\mathrm{T}_3$**

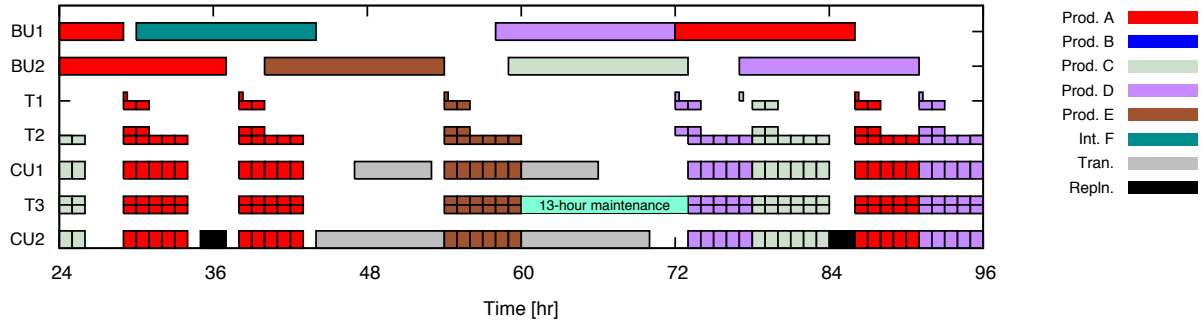
In the third case, a 13-hour maintenance of $\mathrm{T}_3$ needs to be performed in Day 3 from $t = 60$ to 73. This maintenance constraint can be readily specified in the RTN model by letting the upper resource limit of buffer tank $\mathrm{T}_3$ equal zero during corresponding time periods. The maintenance plan becomes known at $t = 24$, the scheduling algorithm is able to take advantage of the knowledge, and design an optimal schedule as shown in Fig. 6.11(c). The most significant change is that the production of Prod. D is brought forward, where the product transitions in $\mathrm{CU}_1$ and $\mathrm{CU}_2$ are carried out while $\mathrm{T}_3$ is not available.




(a) Optimal schedule without process disruptions




(b) Optimal schedule with task delay in $\mathrm{BU}_1$




(c) Optimal schedule with $\mathrm{T}_3$ maintenance

Figure 6.11: Optimal schedules of the mixed plant for Day 2-4 in different scenarios

| Scenario | Opt. obj. | Gap (%) | Var. (Discrete) # | Con. # | Solver | CPU(s) |
|---|---|---|---|---|---|---|
| 1.(Fig. 6.11(a)) | 635.2 | 4.6 | 134,148(4,380) | 140,382 | Gurobi 5.5.0 | 600 |
| 2.(Fig. 6.11(b)) | 634.8 | 4.7 | 134,148(4,380) | 140,382 | Gurobi 5.5.0 | 600 |
| 3.(Fig. 6.11(c)) | 633.5 | 4.9 | 134,148(4,380) | 140,382 | Gurobi 5.5.0 | 600 |

Table 6.2: Model and solution statistics of the reactive cases

Under all three circumstances, the plant is able to fulfill customer orders and maintain product inventory levels, except for Prod. E in the second case (but it is expected to be recovered in the near future). Note that we illustrate the rolling horizon scheme by considering rescheduling per day, but in reality it can be invoked at a much higher frequency. The model size remains the same for all cases. The computation times are $600$ CPU seconds, where the solutions have optimality gaps smaller than $5\%$.

## 6.7 Concluding Remarks

An optimal scheduling formulation based on the extended discrete time RTN representation has been developed. The extended RTN model is versatile enough to capture many detailed features of the plant operation, such as modeling mixed batch/continuous units, product changeovers, and consumable processing capacity replenishment. Moreover, the scheduling model is converted to the state space form to carry out reactive scheduling tasks, working in conjunction with the rolling horizon scheme. The state space model incorporates *lifted* state variables to record task histories, which can be efficiently exploited in defining rescheduling problems and modeling process disruptions. In the case study, we have tested the scheduling formulation in different scenarios, especially in the context of responding to process disruptions. The formulation is able to address these problems as it successfully designs reasonably good schedules within short computation time limits. Such a scheduling model, utilizing a rolling horizon scheme, is desirable to handle practical scheduling problems, where frequent and rapid responses to process disrup-

tions are required. Also, the state space model provides opportunities to conduct stability analysis of the closed-loop scheduling method, analogous to those for MPC algorithms. More importantly, the state space RTN model can also be potentially used in integrated scheduling and control problems, as the lower layer control models are described in a consistent manner.

## 6.8 RTN Model Notation

| *Indices/Sets* | |
|---|---|
| $i/\mathcal{I}$ | tasks |
| $n/\mathcal{N}$ | task extents |
| $o/\mathcal{O}$ | orders |
| $r/\mathcal{R}$ | resources |
| $t/\mathcal{T}$ | time slots |
| $\theta$ | time shift index |
| $\Omega_{o,r}$ | map between order $o$ and resource $r$ |
| *Parameters* | |
| $c_{o,r,t}^{order}$ | order value coefficient |
| $c_i^{tr}$ | cost coefficient of transition tasks |
| $c_r^{inv}$ | inventory penalty coefficients of buffer tanks |
| $c_r^{sl}$ | cost coefficients for violating safety stock levels |
| $d_{i,r,\theta}$ | binary task delay indicator |
| $D_o$ | due date of order $o$ |
| $E_o$ | early date of order $o$ |
| $f_r$ | discount factor of product resource $r$ |
| $H$ | total time horizon length |
| $Q_{o,r}$ | demand of product resource $r$ in order $o$ |
| $\Delta t$ | unit time slot length |
| $V_{i,n}^{max}, V_{i,n}^{min}$ | upper and lower bounds of extent $n$ of task $i$ |
| $\alpha_{i,r,\theta}, \alpha'_{i,r,\theta}$ | discrete task resource limit interaction parameters |
| $\beta_{i,r,\theta}, \beta'_{i,r,\theta}$ | continuous task resource limit interaction parameters |
| $\mu_{i,r,\theta}, \mu'_{i,r,\theta}$ | discrete task resource interaction parameters |
| $\nu_{i,r,\theta}, \nu'_{i,r,\theta}$ | continuous task resource interaction parameters |
| $\tau_i$ | task length |
| *Discrete variables* | |
| $N_{i,t}$ | number of task $i$ starts at time $t$ |
| $\bar{N}_{i,t,\theta}$ | discrete task history state |
| *Continuous variables* | |
| $R_{r,t}$ | excess resource level of resource $r$ at time $t$ |
| $R_{r,t}^{max}, R_{r,t}^{min}$ | upper and lower excess resource level limits of resource $r$ at time $t$ |
| $S_{r,t}^{max}, S_{r,t}^{min}$ | resource slack variables |
| $\xi_{i,n,t}$ | size of batch extent $n$ of task $i$ starts at time $t$ |
| $\bar{\xi}_{i,n,t,\theta}$ | continuous task history state |
| $\Pi_{r,t}$ | external resource transfer of resource $r$ at time $t$ |

# Chapter 7

# Integrated Optimization Strategy

This chapter presents an integrated optimization for industrial polymerization processes, incorporating the developments from the previous chapters on reactor modeling and production scheduling. The method introduce a discrete time formulation for simultaneous optimization of the scheduling and operation decisions. The process is described by the resource task network (RTN) representation coupled with detailed dynamic models. General complications in scheduling and control can be fully represented in this modeling framework, such as customer orders, transfer policy, and requirements on product quality and process safety. The scheduling and operation layers are linked with the task history state variables in the state space RTN model. A tailored generalized Benders decomposition (GBD) algorithm is applied to efficiently solve the resulting large nonconvex mixed-integer nonlinear program by exploring the particular model structure. We apply the integrated optimization approach to a polymerization process that borrows a lot of features from the previous studies. The process has two parallel semi-batch reactors for ring-opening polymerization, continuous storage tanks, and purification units. The two polymerization reactors share cooling utility from the same source, and the utility price depends on the consumption rate. The optimization objective is to design the pro-

cess schedule and reactor control policies simultaneously to maximize the overall process profit. The case study results suggest improvements in plant profitability for the integrated approach, in contrast to the conventional approach, where recipes of the polymerization tasks are individually optimized and the interactions among process units are overlooked.

## 7.1 Model Formulation

We first demonstrate the key equations for process scheduling based on the RTN representation, and then deal with dynamic process models. We use the state space RTN model in Chapter 6 and the moment model for polymerization reactors in Chapter 5.

### 7.1.1 Scheduling with RTN

The linear state space RTN model equations can be carried over to the integrated problem, with a few modification. An important difference is the task mode index $m$, which is added to enable different operation strategies for the same task. For example, the task length can vary by modes, product quality constraints can be set to different threshold values, etc. This provides opportunities for dynamic optimization to have influence on the schedule design by altering execution modes, though the resource transform relationship remains the same for all modes of the same task from the scheduling point of view. Also, binary task assignment variables are used instead of integer ones, as one occurrence of a task can be operated differently than another.

First, the resource balance equation is as follows:

$$R_{r,t} = R_{r,t-1} + \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} \sum_{\theta=0}^{\tau_{i,m}} \mu_{i,m,r,\theta} \bar{w}_{i,m,t,\theta} + \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} \sum_{n\in\mathcal{N}} \sum_{\theta=0}^{\tau_{i,m}} \nu_{i,m,n,r,\theta} \bar{\xi}_{i,m,n,t,\theta} + \Pi_{r,t}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}. \tag{7.1}$$

For variables names, $R$ is the resource level, $\bar{w}$ and $\bar{\xi}$ are the discrete and continuous task history states, $\mu$ and $\nu$ are the corresponding task resource interaction parameters, and $\Pi$ is used to represent external transfer events. For the indices, $r$ is resource, $t$ is time, $i$ is task, $m$ is task mode, $n$ is task extent, and $\theta$ is a dummy index used for time shift. Next, the lifting equations for the task assignment and batch sizing variables are obtained:

$$\bar{w}_{i,m,t,\theta} \triangleq w_{i,m,t-\theta}, \quad \forall i \in \mathcal{I}, m \in \mathcal{M}, t \in \mathcal{T}, 0 \leqslant \theta \leqslant \tau_{i,m}; \tag{7.2a}$$

$$\bar{\xi}_{i,m,n,t,\theta} \triangleq \xi_{i,m,n,t-\theta}, \quad \forall i \in \mathcal{I}, m \in \mathcal{M}, n \in \mathcal{N}, t \in \mathcal{T}, 0 \leqslant \theta \leqslant \tau_{i,m}. \tag{7.2b}$$

The change of the task history states over time is described by the state evolution equations:

$$\bar{w}_{i,m,t,\theta} = \bar{w}_{i,m,t-1,\theta-1}, \quad \forall i \in \mathcal{I}, m \in \mathcal{M}, t \in \mathcal{T}, 1 \leqslant \theta \leqslant \tau_{i,m}; \tag{7.3a}$$

$$\bar{\xi}_{i,m,n,t,\theta} = \bar{\xi}_{i,m,n,t-1,\theta-1}, \quad \forall i \in \mathcal{I}, m \in \mathcal{M}, n \in \mathcal{N}, t \in \mathcal{T}, 1 \leqslant \theta \leqslant \tau_{i,m}. \tag{7.3b}$$

For resource limit balance relations:

$$R^{max}_{r,t} = R^{max}_{r,t-1} + \sum_{i \in \mathcal{I}} \sum_{m \in \mathcal{M}} \sum_{\theta=0}^{\tau_{i,m}} \alpha_{i,m,r,\theta} \bar{w}_{i,m,t,\theta} + \sum_{i \in \mathcal{I}} \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} \sum_{\theta=0}^{\tau_{i,m}} \beta_{i,m,n,r,\theta} \bar{\xi}_{i,m,n,t,\theta}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}; \tag{7.4a}$$

$$R^{min}_{r,t} = R^{min}_{r,t-1} + \sum_{i \in \mathcal{I}} \sum_{m \in \mathcal{M}} \sum_{\theta=0}^{\tau_{i,m}} \alpha'_{i,m,r,\theta} \bar{w}_{i,m,t,\theta} + \sum_{i \in \mathcal{I}} \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} \sum_{\theta=0}^{\tau_{i,m}} \beta'_{i,m,n,r,\theta} \bar{\xi}_{i,m,n,t,\theta}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}. \tag{7.4b}$$

Lastly, inequality constraints for resource levels and task extent sizes:

$$R^{min}_{r,t} \leq R_{r,t} \leq R^{max}_{r,t}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}. \tag{7.5}$$

$$V^{min}_{i,m,n} \bar{w}_{i,m,t,\theta} \leq \bar{\xi}_{i,m,n,t,\theta} \leq V^{max}_{i,m,n} \bar{w}_{i,m,t,\theta}, \quad \forall i \in \mathcal{I}, m \in \mathcal{M}, n \in \mathcal{N}, t \in \mathcal{T}, 0 \leqslant \theta \leqslant \tau_{i,m}. \tag{7.6}$$

### 7.1.2 Operation Optimization with Reactor Models

Moment models are used to model the two polymerization reactors. Here, we consider homopolymer products that are polymerized via ring-opening. For detailed equations please refer to Appendix D. Nevertheless, the reactor models are DAE systems that can be can be written in a generic form as follows:

$$\dot{z} = f(z(s), y(s), u(s)), \quad z(0) = z_0; \tag{7.7a}$$

$$g(z(s), y(s), u(s)) = 0. \tag{7.7b}$$

The differential equations are defined for the differential states $z$ in semi-explicit form, and the algebraic states are $y$. The initial condition of the differential states is known as $z_0$. Process control variables are $u$. We use $s$ to denote the continuous time coordinate in contrast to $t$ that is used for the discrete time grid. Without loss of generality, process constraints for optimization are expressed as:

$$h(z(s), y(s), u(s)) \leqslant 0. \tag{7.8}$$

### 7.1.3 Integrated Formulation

The integrated formulation ties the RTN scheduling model and process dynamic models together. First, time representation needs to be considered, as a means to coordinate scheduling time slots, finite elements, and collocation points. A unified time grid is defined as shown in Fig. 7.1. In this example, the entire scheduling time horizon $t \in [0, H]$ is filled with $H$ discrete time slots of unit length. A time slot contains two finite elements with three collocation points inside each element. The arrangement is adjustable for specific applications, e.g., the number of finite elements in a time slot and the type and number of collocation points used. Note that the starting point of a scheduling time slot is

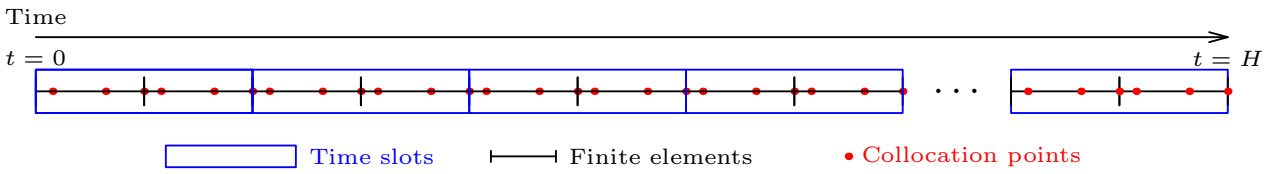



Figure 7.1: Time representation of the integrated formulation

aligned with a finite element. By this means, task switching events can be simultaneously addressed in both the scheduling and dynamic control equations.

Task switching events are described by the task assignment/history variables in the RTN scheduling model. In the lower layer dynamic optimization problem, process dynamic models are influenced by switching tasks. This may correspond to changes in the model parameters or structure, as well as different specifications on the process constraints. Therefore, the dynamic system and constraints are also functions of the task history states in the integrated model. In a time slot $t$, we obtain

$$\dot{z}_t = f(z_t, y_t, u_t, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}), \quad z_t^0 = \mathcal{Z}(z_{t-1}^f, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}); \tag{7.9a}$$

$$g(z_t, y_t, u_t, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) = 0; \tag{7.9b}$$

$$h(z_t, y_t, u_t, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) \leqslant 0. \tag{7.9c}$$

The equations are in the continuous form for notational convenience, and the discretized version can be obtained through the simultaneous collocation method. Note that the states and controls are defined with respect to the entire process such that they do not have indices for task or task mode dependency. Moreover, the initial conditions of the differential states are no longer known parameters but a function of its value at the end of the previous slot ($z_{t-1}^f$) and task assignment variables. If no task switch occurs, $z_t^0$ and $z_{t-1}^f$ are equal to ensure the continuity of the differential states; otherwise, sudden jumps may occur due to the start or finish of operations. This discrete switching behavior is described by $\mathcal{Z}(\cdot)$, which is a balance equation of $z_t$ across the discrete time slot boundaries, similar

to the resource balance equation (Eq (7.1)).

For the objective function, profit maximization is a common choice for the integrated scheduling and dynamic optimization problem, since it accounts for the trade-offs between product sales and manufacturing costs in a natural way. Product sales are decided by the scheduling level decisions such as batch numbers, while manufacturing costs are calculated using the detailed dynamic models.

To summarize, the integrated formulation for scheduling and dynamic optimization can be described as follows:

$$
\begin{aligned}
\max \quad & \underbrace{\Phi(R_{r,t}, z_t, y_t, u_t, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta})}_{\text{Overall profit}} = \underbrace{\Phi^{sch}(R_{r,t}, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta})}_{\text{Product sales}} - \underbrace{\Phi^{dyn}(z_t, y_t, u_t, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta})}_{\text{Manufacturing cost}} \\
s.t. \quad & F^{sch}(R_{r,t}, R^{min}_{r,t}, R^{max}_{r,t}, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) = 0 \quad \text{Scheduling equalities: Eqs. (7.1), (7.3), (7.4)} \\
& G^{sch}(R_{r,t}, R^{min}_{r,t}, R^{max}_{r,t}, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) \leqslant 0 \quad \text{Scheduling inequalities: Eqs. (7.5), (7.6)} \\
& F^{dyn}(z_t, y_t, u_t, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) = 0 \quad \text{Dynamic equalities: Eqs. (7.9a), (7.9b) (discretized)} \\
& G^{dyn}(z_t, y_t, u_t, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) \leqslant 0 \quad \text{Dynamic inequalities: Eq. (7.9c) (discretized)}
\end{aligned}
\tag{7.10}
$$

Here, the overall objective function $\Phi(\cdot)$ is written as a linear combination of two sub-objectives $\Phi^{sch}$ and $\Phi^{dyn}$, depending on the scheduling and dynamic optimization variables respectively. Also, the constraints can be separated into scheduling and dynamic optimization groups.

## 7.2 Solution Strategy

The integrated formulation is posed as a nonconvex MINLP problem, and the direct solution of which is time-consuming or even computationally intractable for large-scale

problems. However, the model structure (Eq. (7.10)) can be efficiently explored with decomposition algorithms. The scheduling equations and dynamic models are only linked by the task history state variables, and the remaining parts are isolated from each other. The GBD method is well suited for this type of problem. In this method, the original model is decoupled to a primal problem and a master problem. In the primal problem, the linking variables (also termed as complicating variables) are temporarily fixed to some value such that the remaining parameterized problem has a more tractable size and structure. The master problem is a projection of the original problem to a restricted variable space, i.e., the variables in the subproblem (except for the complicating variables) are not explicitly included in the master problem. Alternatively, the master problem includes cutting planes which are constructed from the solution and dual information of the primal problem. The optimal solution of the original problem is bounded by the best primal and master problem solutions. For this section we discuss the main components of the GBD method in the context of integrated scheduling and dynamic optimization.

### 7.2.1 Primal Problem

The primal problem for the integrated formulation is the dynamic optimization problem (Eq. (7.9)) with fixed task history states. Because all discrete variables are fixed, the problem is reduced to a continuous NLP:

$$\begin{aligned} \min_{u_t} \quad & \Phi^{dyn}(z_t, y_t, u_t, \bar{w}^p_{i,m,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) \\ s.t. \quad & F^{dyn}(z_t, y_t, u_t, \bar{w}^p_{i,m,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) = 0; \\ & G^{dyn}(z_t, y_t, u_t, \bar{w}^p_{i,m,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) \leqslant 0. \end{aligned} \tag{7.11}$$

The superscript $p$ of $\bar{w}$ and $\bar{\xi}$ indicates that they are fixed parameters. The objective function considers only the manufacturing cost $\Phi^{dyn}(\cdot)$, and it is rewritten as a minimization problem following the convention of the GBD method. Assuming the primal problem

is solved to optimality by NLP algorithms with given history state values, the Lagrange function $\mathcal{L}$ can be constructed as follows:

$$\begin{aligned}\mathcal{L}(\bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) = \quad & \Phi^{dyn*} + \lambda^{*\mathrm{T}} F^{dyn}(z_t^*, y_t^*, u_t^*, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) \\ & + \xi^{*\mathrm{T}} G^{dyn}(z_t^*, y_t^*, u_t^*, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}); \\ & 0 \leqslant \xi^* \perp G^{dyn}(z_t^*, y_t^*, u_t^*, \bar{w}^p_{i,m,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) \geqslant 0.\end{aligned} \tag{7.12}$$

Here, $\lambda$ and $\xi$ are the vectors of optimal Lagrange multipliers for $F^{dyn}(\cdot)$ and $G^{dyn}(\cdot)$ obtained at the optimal solution, and the optimal objective and variable values are noted with $^*$. The complementarity condition of the inequalities needs is satisfied between $\xi^*$ and $G^{dyn}(z_t^*, y_t^*, u_t^*, \bar{w}^p_{i,m,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta})$. The structure of the Lagrange function can be complex as the number of equations in $F^{dyn}(\cdot)$ and $G^{dyn}(\cdot)$ are often very large after collocation. An alternative approach that constructs a more concise Lagrange function is to introduce duplicated complicating variables in Eq. (7.11):

$$\begin{aligned}\min_{u_t} \quad & \Phi^{dyn}(z_t, y_t, u_t, \bar{w}^p_{i,m,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) \\ & F^{dyn}(z_t, y_t, u_t, \bar{w}^p_{i,m,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) = 0, \\ & G^{dyn}(z_t, y_t, u_t, \bar{w}^p_{i,m,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) \leqslant 0, \\ s.t. \quad & \bar{w}^p_{i,m,t,\theta} = \bar{w}^d_{i,m,t,\theta}, \\ & \bar{\xi}^p_{i,m,n,t,\theta} = \bar{\xi}^d_{i,m,n,t,\theta}.\end{aligned} \tag{7.13}$$

To derive the Lagrange function, we obtain

$$\begin{aligned}\mathcal{L}(\bar{w}^d_{i,m,t,\theta}, \bar{\xi}^d_{i,m,n,t,\theta}) = \quad & \Phi^{dyn*} + \lambda^{*\mathrm{T}} F^{dyn}(z_t^*, y_t^*, u_t^*, \bar{w}^p_{i,n,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) \\ & + \xi^{*\mathrm{T}} G^{dyn}(z_t^*, y_t^*, u_t^*, \bar{w}^p_{i,m,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) \\ & + \sigma^{*\mathrm{T}}(\bar{w}^p_{i,m,t,\theta} - \bar{w}^d_{i,m,t,\theta}) + \omega^{*\mathrm{T}}(\bar{\xi}^p_{i,m,n,t,\theta} - \bar{\xi}^d_{i,m,n,t,\theta}) \\ = \quad & \Phi^{dyn*} + \sigma^{*\mathrm{T}}(\bar{w}^p_{i,m,t,\theta} - \bar{w}^d_{i,m,t,\theta}) + \omega^{*\mathrm{T}}(\bar{\xi}^p_{i,m,n,t,\theta} - \bar{\xi}^d_{i,m,n,t,\theta}).\end{aligned} \tag{7.14}$$

Here, $\sigma$ and $\omega$ are the corresponding Lagrange multipliers for the duplication equations. The simplification is due to the fact that the discretized dynamic equations are not a function of the duplicated variables, and

$$\lambda^{*\mathrm{T}} F^{dyn}(z_t^*, y_t^*, u_t^*, \bar{w}^p_{i,n,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) = 0$$
$$\xi^{*\mathrm{T}} G^{dyn}(z_t^*, y_t^*, u_t^*, \bar{w}^p_{i,m,t,\theta}, \bar{\xi}^p_{i,m,n,t,\theta}) = 0$$

at the primal optimum. This concise Lagrange function is equivalent to the one in Eq. (7.12) and computationally more favorable [36]. The Lagrange function is used to form the cutting plane constraints (a.k.a. Benders cuts) in the master problem.

### 7.2.2 Master Problem

The master problem contains the scheduling equations used in the linear RTN model (Eqs. (7.1), (7.3), (7.4), (7.5), (7.6)), as well as the cutting plane constraints, which can be stated as:

$$\min \quad -\Phi^{sch}(R_{r,t}, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) + \eta \tag{7.15a}$$

$$\begin{aligned} s.t. \quad & F^{sch}(\bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}, R_{r,t}, R^{min}_{r,t}, R^{max}_{r,t}) = 0, \\ & G^{sch}(\bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}, R_{r,t}, R^{min}_{r,t}, R^{max}_{r,t}) \leqslant 0, \\ & \mathcal{L}(\bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) \leqslant \eta. \end{aligned} \tag{7.15b}$$

There is an infinite number of cutting planes that can be constructed by the Lagrange function. A typical strategy is to replace the master problem with its relaxation, where

only a finite collection of the cutting planes are included:

$$\min \quad -\Phi^{sch}(R_{r,t}, \bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) + \eta \tag{7.16a}$$

$$s.t. \quad \begin{aligned} & F^{sch}(\bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}, R_{r,t}, R_{r,t}^{min}, R_{r,t}^{max}) = 0, \\ & G^{sch}(\bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}, R_{r,t}, R_{r,t}^{min}, R_{r,t}^{max}) \leqslant 0, \\ & \mathcal{L}^{b}(\bar{w}_{i,m,t,\theta}, \bar{\xi}_{i,m,n,t,\theta}) \leqslant \eta, \quad b \in \mathcal{B}. \end{aligned} \tag{7.16b}$$

Here, the set of valid cutting planes are denoted by $b \in \mathcal{B}$. The cuts can be collected at different primal optimal solutions.

### 7.2.3 GBD Algorithm

The GBD method applies an iterative procedure to obtain the optimal solution of the original problem, where the primal and relaxed master problems are solved in a loop, as depicted in Fig. 7.2. The algorithm often starts with solving the primal problem with the complicating variables fixed to an initial point, and the master problem is then constructed with the obtained Benders cut at the primal optimum. The master problem is solved to update the complicating variable, and triggers the next GBD iteration. The primal solution gives the upper bound of the optimal objective value, while the relaxed master problem calculates the lower bound. As Benders cuts are accumulated in the master problem, the lower bound is non-decreasing through the iterations. The algorithm converges when the two bounds fall in a close neighborhood specified by optimality tolerances [165]. It is worth noting that global optimality is only guaranteed when convexity condition holds in both the primal and master problems. This is usually not satisfied by the integrated scheduling and dynamic optimization problem. Also, integer cuts can be

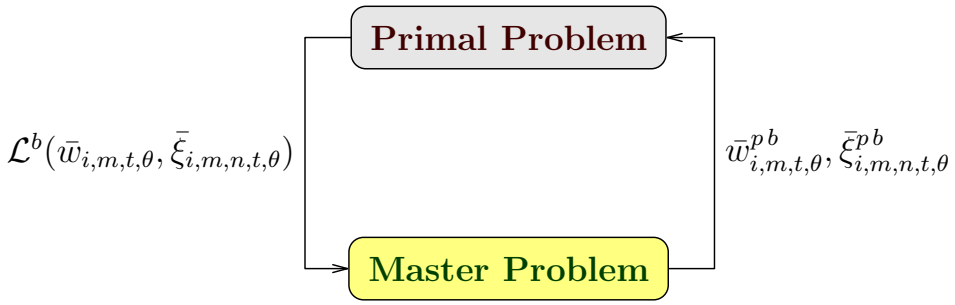



Figure 7.2: GBD iteration scheme

added to the relaxed master problem to avoid cyclic integer assignments:

$$\begin{aligned} &\sum_{\mathcal{W}_1^b} \bar{w}^b_{i,m,t,0} - \sum_{\mathcal{W}_0^b} \bar{w}^b_{i,m,t,0} \leqslant |\mathcal{W}_1^b| - 1; \\ &\{\bar{w}^b_{i,m,t,0} \in \mathcal{W}_1^b | \bar{w}^b_{i,m,t,0} = 1\}, \{\bar{w}^b_{i,m,t,0} \in \mathcal{W}_0^b | \bar{w}^b_{i,m,t,0} = 0\}. \end{aligned} \tag{7.17}$$

The cut are defined with the binary task history state variables $\bar{w}^b_{i,n,t,0}$. An integer cut obtained at iteration $b$ is added to the next master problem at $b+1$.

## 7.3 Case Study

We illustrate the discrete time integrated formulation with a polymerization process. The process flowsheet is shown in Fig. 7.3. The plant has two polymerization reactors (Rxr1, Rxr2) of identical processing capacity in parallel, and they both connect to the downstream buffer tank (T) followed by the purification unit (PU). Raw polymer products are made in the reactors, and PU removes the catalyst in the raw products to give final products. The polymerization reactors are semi-batch units, and both the tank and PU are continuous units. Three types of products are made from the process (A, B, C) with different specs on molecular weight (MW) and byproduct ratio. Rigorous dynamic models are developed for the polymerization reactors, while the other continuous units are modeled by the linear resource balance in the RTN representation. For details on the reactor

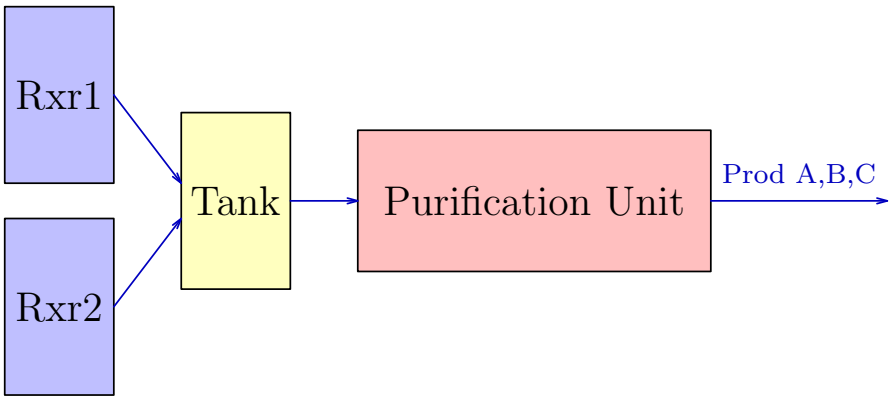



Figure 7.3: Process flowsheet of the polymerization process

model equations and process RTN representation please refer to the Appendix D.

The polymerization reactors demand large amounts of cooling capacity as the reactions are highly exothermic. An interesting feature of the plant is that the two reactors share cooling utility from the same source, and the utility price $p$ is a function of its consumption rate $r$:

$$p = \begin{cases} 1, & 0 \leqslant r \leqslant r_0; \\ 1 + \frac{1}{r_0}(r - r_0), & r_0 \leqslant r \leqslant 3r_0. \end{cases} \tag{7.18}$$

The utility price equals one money unit (MU) when the total utility consumption rate of the reactors is below a basis value $r_0$, and the price rises linearly beyond $r_0$ till the maximum rate $3r_0$, where the price reaches $3$ MU. Other complexities come from both the scheduling and dynamic optimization decisions. For scheduling considerations, Product A is made-to-order while Product B and C are made-to-stock. The optimized schedule is required to produce the exact amount of Product A according to customer orders, while extra Product B and C can be made in addition to ordered amounts. Product A and C are in the same family and Product B is an orphan product. Equipment cleaning is required for PU during transition between different product families. Also, PU needs to carry out *off-line* maintenance after processing a certain amount of raw polymers. In other words, the processing capacity is consumable, and the maintenance can replenish it after it drops

below a threshold value. Finally, mixing of different products is not allowed in any units. For the reactors, a one-hour set-up time is needed before starting any polymerization operations. Main constraints considered are the number-average MW, byproduct ratio, unreacted monomer concentration, and maximum cooling duty. Other minor constraints are also included such as limits on the variation of reactor temperature.

The decision variables are the task assignments for scheduling, and the control strategy of the reactors that include the temperature trajectory and the monomer feed policy. There are two modes for the polymerization of Product B and C with different task lengths but the same quality specs. Product A has only one polymerization mode. PU has a maximum processing rate that is product dependent. The goal of the integrated optimization is to maximize the plant profit within a 3-day scheduling time horizon, which is calculated by the product sales minus the total manufacturing cost. The cost consists of the cooling utility cost, fixed cost for polymerization operations and PU maintenance, as well as penalties for monomer flow rate and temperature fluctuations.

We solve the integrated optimization problem with the GBD decomposition approach. The primal problem is defined as dynamic optimization of the polymerization operations in the two reactors to minimize the manufacturing cost, and the master problem deals with the complete process as a linear scheduling problem without the dynamic reactor models. The complicating variables are the task history states associated with the polymerization reactors. We use one-hour scheduling time slots, and only one finite element is included for a time slot. Three Radau collocation points are employed in a finite element. The model is built in GAMS, and we use Gurobi to solve the master MILP problems and Conopt to solve the primal NLP problems.

We compare the integrated optimization approach with the conventional approach, where polymerization recipes are optimized individually and the schedule is determined with the fixed recipe parameters. This requires solving separate dynamic optimization prob-

lems for recipe optimization and a discrete optimization problem for schedule design. The scheduling problem is a mixed-integer quadratic program (MIQP) because it includes a quadratic term for the cooling utility cost in its objective function, despite all model constraints are linear. The model and solution statistics for the conventional approach is given in Tab. 7.1. Five recipe optimization problems are solved for the five polymerization modes (See Tab.D.1), where the objective function is the manufacturing cost. All five cases result in small scale NLPs that are solved within seconds. The reactor model has the same structure for the five cases (see the Appendix D), but model parameters may vary such as the initial charge condition, kinetic constants, and product specs. The difference in model sizes is mainly due to the number of finite elements used in each problem. Longer tasks require more finite elements, as one-hour element is used in this case study. The scheduling problem are solved within 1200 seconds limit and the optimality gap $\sim 5\%$. The MIQP problem is difficult to solve not only because its large size but also the degenerate nature of scheduling problems, where many different schedules (typically different in task timing) give the same objective value [166]. Note that the number of continuous variables in these RTN scheduling problems are significantly larger than that in the conventional RTN models. This is due to the inclusion of the lifted task state variables.

For the integrated formulation, the GBD algorithm is able to solve the problem with three iterations, and model and solution statistics are shown in Tab. 7.2. The algorithm starts with the so-called zeroth primal problem, which is the first dynamic optimization problem with given task states. These initial values of the task states are obtained at the optimal solution of the conventional scheduling problem (MIQP). Then the GBD algorithm iterates between the master MILP problem and the primal NLP problem. The major percentage of CPU time is spent on the primal problems, given that NLPs suffer from their large sizes and degenerate structure. It is worthing noting that the NLP problems need careful initialization in order to be solved successfully at each iteration. The fixed recipes can be used to carry out simulation once the schedule is determined, and this offers good

| Recipe optimization | | | | | |
|---|---|---|---|---|---|
| Product (mode) | Type | Model solution | | Model size | |
| | | Obj. (MU) | CPU time (s) | Var. # | Cons . # |
| A | NLP | 5.04 | 3.3 | 1200 | 1239 |
| B(1) | NLP | 7.46 | 3.7 | 832 | 859 |
| B(2) | NLP | 6.99 | 2.5 | 924 | 954 |
| C(1) | NLP | 10.86 | 4.0 | 648 | 669 |
| C(2) | NLP | 8.76 | 2.4 | 740 | 764 |
| Schedule optimization | | | | | |
| | Type | Model solution | | Model size | |
| | | Obj. (MU) | CPU time (s) | Var. #(/dis.) | Cons . # |
| | MIQP | 82.56 | 1200.0 | 30779(1825) | 30823 |

Table 7.1: Model and solution statistics for the case study: conventional approach

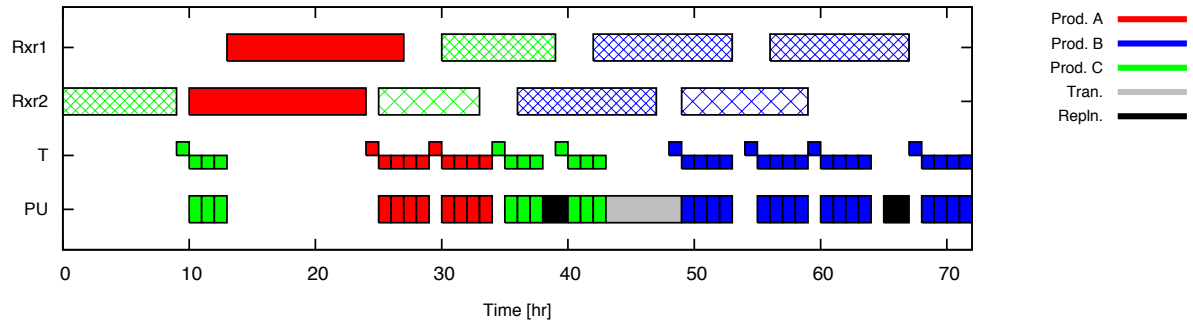
| Iteration | Type | Model solution | | Model size | | GBD algorithm | | |
|---|---|---|---|---|---|---|---|---|
| | | Obj.(MU) | Time (s) | Var.#(/dis.) | Cons .# | Upper | Lower | Gap (%) |
| 0th primal | NLP | 68.17 | 53.6 | 14820 | 16570 | | 85.16 | |
| 1st master | MILP | 120.94 | 4.8 | 30561(1825) | 30666 | 120.94 | 85.16 | 29.58 |
| 1st primal | NLP | 72.23 | 199.2 | 14820 | 16570 | 120.94 | 89.04 | 26.38 |
| 2nd master | MILP | 104.22 | 10.4 | 30561(1825) | 30668 | 104.22 | 89.04 | 14.57 |
| 2nd primal | NLP | 79.10 | 120.4 | 14820 | 16570 | 104.22 | 92.26 | 11.48 |
| 3rd master | MILP | 92.26 | 4.5 | 30561(1825) | 30670 | 92.26 | 92.26 | 0.00 |

Table 7.2: Model and solution statistics for the case study: integrated approach

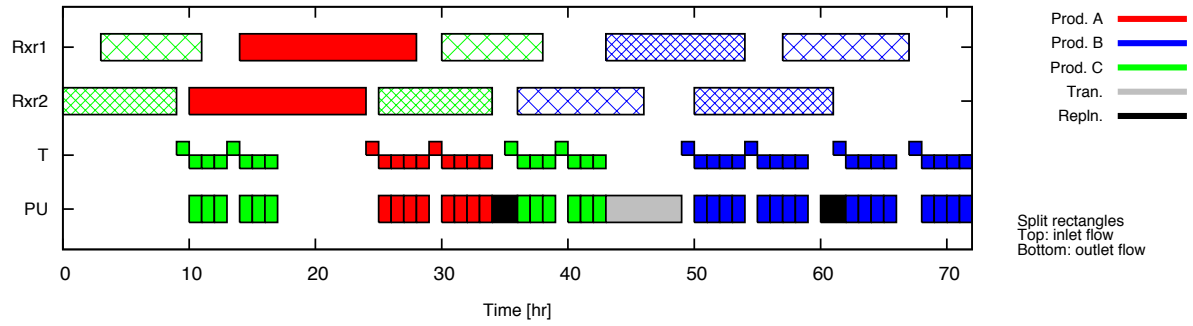
starting points for the primal NLPs. The master problems are solved to zero optimality gap, where the solution speed is fast in contrast to the MIQP problem for the conventional approach, although the size of models are very close. This is due to the inclusion of the Benders cuts that helps to eliminate the degeneracy of the scheduling problem. The optimizer is able to evaluate schedules with different timings with the assistance of the multiplier values for the task history states ($\sigma$ and $\omega$ in Eq. (7.14)). As a maximization problem, the upper and lower bounds of the optimal objective are given by the primal and master problems, respectively. The optimality gap is reduced to zero after the third master problem.

The maximal plant profit is $92.26$ MU for the integrated optimization approach, which is $10.5\%$ higher than that of the conventional approach. The optimized schedules are shown in Fig. 7.4 in comparison. Task rectangles are colored to denote different products. Also, different polymerization tasks modes have different fill patterns. For the buffer tank, the upper small rectangles represent the material transfer operations that load the reactor products to the buffer tank, while the lower rectangles represent the outgoing flows from the tank to PU. It is worth noting that the reactors can be used as temporary storage units to hold batches when the buffer tank is not immediately available. In fact, polymerization carries on in the interim, because the catalyst is still effective as the reactor temperature is maintained in the allowable range. PU processes different products, carries out transition cleaning, and off-line maintenance to replenish its processing capacity. In the schedule of the conventional approach (Fig. 7.4(a)), two batches of Product A, four batches of Product B, and three batches of Product C are produced. Also, different task modes are observed for Product B and C. One additional batch of Product C is observed in the schedule of the integrated approach (Fig. 7.4(b)). This is because the polymerization recipes are fixed in the conventional approach, and running two batches of Product C in parallel is undesirable that leads to very expensive utility prices (see Fig. 7.9). However, the operating flexibility can be efficiently explored in the integrated approach, such that the utility price can be brought down to a level where the additional batch is profitable. The main disadvantage of the conventional method is that it overlooks the interaction between the two reactors. More specifically, dynamic optimization for individual recipes is performed with respect to a single reactor, while the actual utility price is determined by the summation of the utility consumption rates of the two reactors. However, this synergy cannot be considered by the conventional approach since no production schedules are given in advance.

The inventory profiles of the intermediate tank is shown in Fig. 7.5(a) and the process rates of PU is shown in Fig. 7.5(b). For the details of reactor operations, we show the



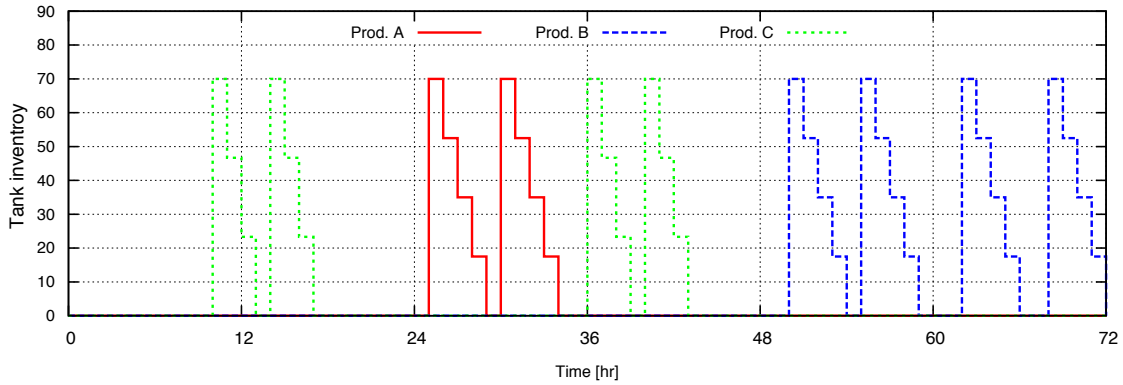

(a) Conventional approach



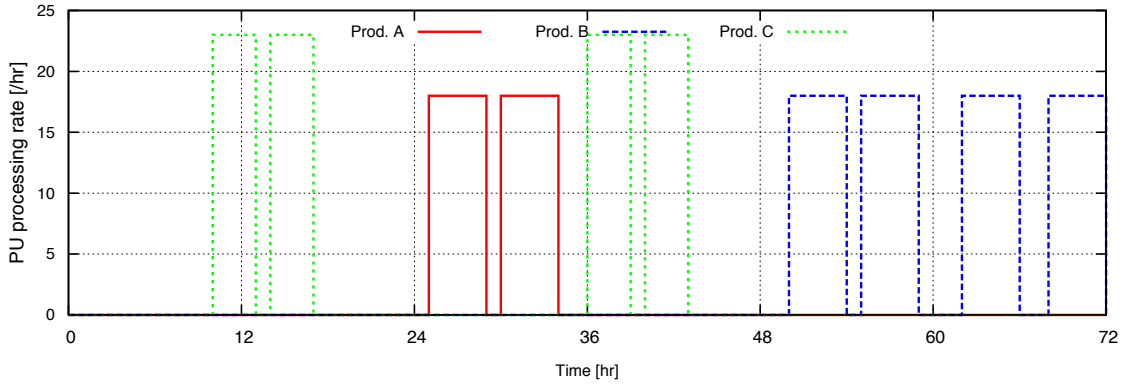

(b) Integrated approach

Figure 7.4: Optimal production schedules

results of the integrated formulation in comparison with a base case. The base case adopts the optimal schedule from the integrated optimization problem (Fig. 7.4(a)), but fixes the reactor control profiles to the recipe values obtained in the conventional approach. Fig. 7.6 gives the optimal reactor temperature and monomer feeding profiles over the entire time horizon. The optimal solution from the integrated formulation is depicted in red solid lines and the blue dotted lines sketch the base case profiles. The plotted temperature data is in Celsius and calibrated with respect a basis level value $T_b$; and the monomer flow rates are scaled. Since the variation of reactor temperature is penalized, gradual transitions in the temperature curves are observed from one batch to another. The monomer is allowed to enter the reactors one hour after the polymerization tasks start, due to the set-up requirements. It is worth noting that the feed profiles tend to minimize the overlaps between two reactors. Particularly for the integrated solution, it gives flow shapes such that the summation of the rates of the two reactors are reduced compared to




(a) Optimal tank inventory

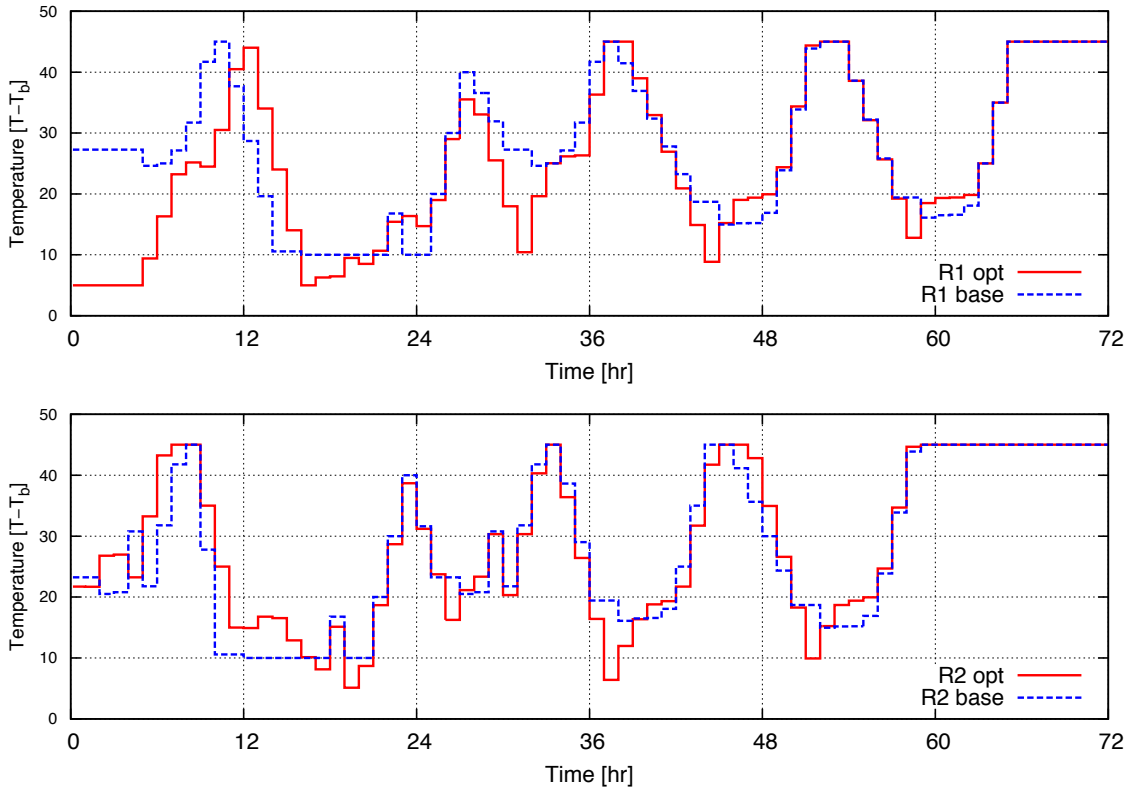



(b) Optimal purification rate

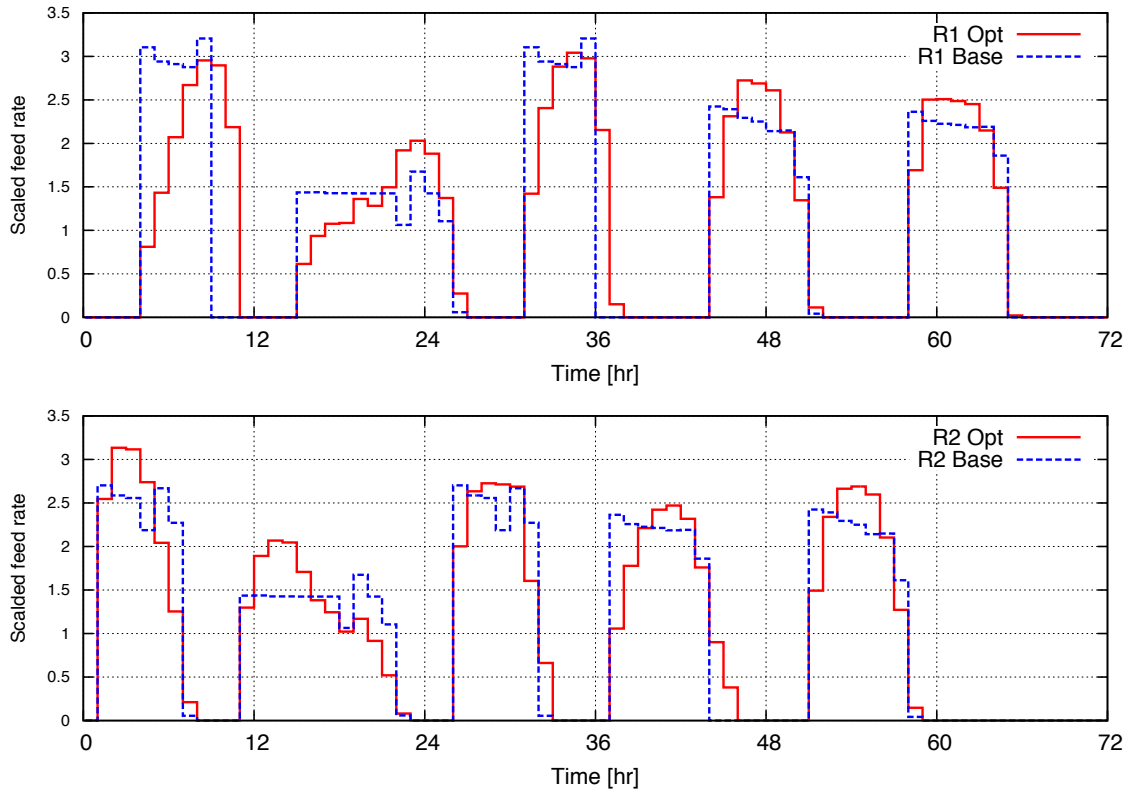
Figure 7.5: Operating profiles for the downstream units

the base case. This is due to the fact that the feed speed is positively related to the heat generation rates in the reactors, and therefore, small total feed rates lead to lower cooling duty which comes with lower utility prices.

The maximum cooling capacity and product quality constraints (specs on the byproduct ratio, number-average MW, and unreacted monomer concentrations) are important constraints that should be satisfied for all polymerization batches. We show these constraint profiles of Rxr1. The utility consumption profiles are plotted together with the maximum cooling capacity limits for both cases in Fig. 7.7. We measure the rate with $r_0$ as the base unit. The cooling capacity limit is depended on the temperature of the reactors so that it varies in time. The constraint on cooling capacity is active for most of the operation periods. In Fig. 7.8, the byproduct ratio should stay below its spec limits, while the polymer




(a) Optimal reactor temperature profiles




(b) Optimal monomer feed profiles

Figure 7.6: Comparison of the optimal control recipes

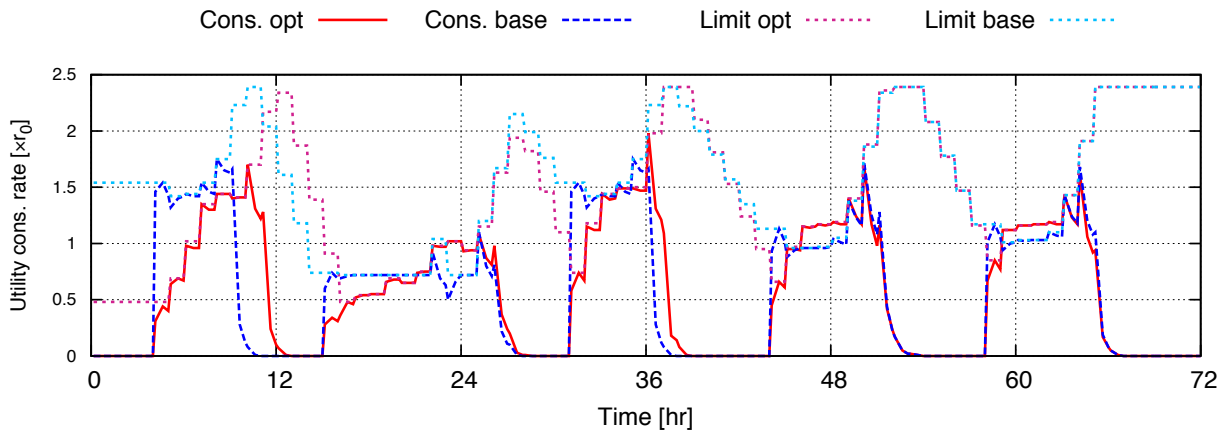



Figure 7.7: Process constraint on the maximum cooling rate

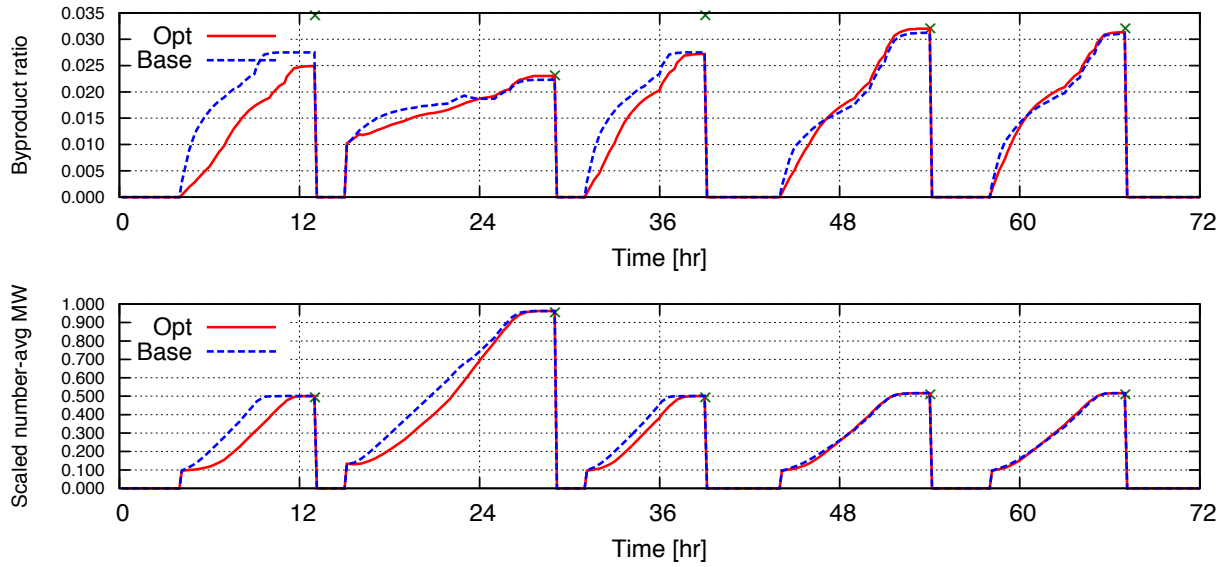



Figure 7.8: Process constraint on the product quality indices

MW should be no less at the end of polymerization. In the plots, the target spec values are noted with cross markers. The byproduct ratio and MW have product dependent target values. The unreacted monomer concentration also reaches desired levels for all polymerization runs. These quality constraints are only effective at the grid points when raw polymers are transferred to the buffer tank, and otherwise relaxed by using big M terms in the model formulation ($\mathcal{Z}(\cdot)$ in Eq. (7.9a)).

In Fig. 7.9, the cooling utility price, total consumption, and total cost are plotted over time. In the price plot, high price periods occur in the first half of the time horizon,

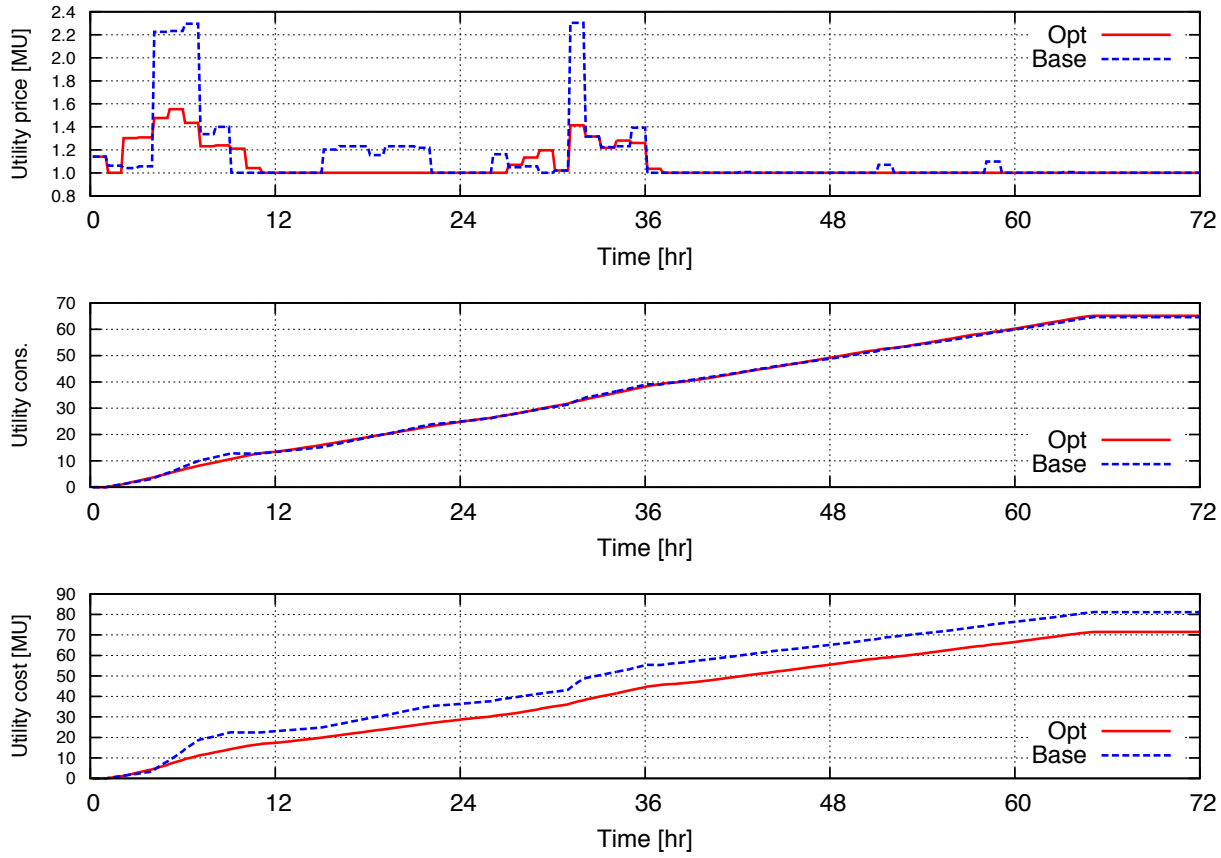



Figure 7.9: Comparison of the cooling utility consumption and cost

especially at the beginning when polymerizing Product C. The optimized recipe is able to reduce the peak values of the high price segments and also their occurrence frequency. The middle graph shows the accumulated total utility consumption of the two reactors. The amounts of consumption are equal for the two cases at the final time, as the same amount of products is made. However, we see a lower cost for the integrated solution in the utility cost curve because it takes advantage of lower prices.

## 7.4 Concluding Remarks

We have developed a general framework for the integration of production scheduling and dynamic optimization decision-making. The method combines the state space RTN rep-

resentation with rigorous process dynamic models to formulate the integration problem as MIDOs. The scheduling and control layers are linked through the lifted task history state variables, which provide opportunities to apply the GBD decomposition method for efficient problem solution. The GBD algorithm decouples the problem into the linear master scheduling problem and continuous primal dynamic optimization problem. The two problems are solved in a loop and Benders cuts are derived to close the optimality gap between the two. We have applied our method to a polymerization process with networked structure and multiple polymer products. The polymerization reactors are described by first-principles models. The objective is to maximize the plant profitability. The cooling utility cost brings in interactions between the two reactors, and we show the integrated approach has advantages over the conventional approach, where the control recipes are determined separately.

# Chapter 8

# Conclusions

Model-based optimization strategies are powerful tools that improve chemical plant operations. Integrated optimization methods offer further advantages by fully exploring the degrees of freedoms in decision-making. In particular, we have investigated the integration of process scheduling and dynamic optimization algorithms that links between discrete tactical and dynamic operational decisions. The industrial case studies are centered on ring-opening polymerization processes, and we address the problems in reactor modeling, polymerization recipe optimization, production scheduling, and integrated scheduling and dynamic optimization in a sequential way. This chapter concludes the thesis work by summarizing the main contributions and recommended future directions.

## 8.1 Thesis Summary and Contributions

In Chapter 3, we have developed an integrated optimization formulation for batch processes with general network structures. The method uses the unit-specific continuous time representation, and the state equipment network (SEN) representation is carried over to scheduling problems. The SEN representation method well accommodates dy-

namic process models with switching behaviors. The overall problem is posed as a mixed logic dynamic optimization (MLDO) problem to maximize plant profits in a given scheduling time horizon. A solution procedure for MLDO problems is discussed. The big M method is employed to reformulate logic disjunctions to mixed-integer constraints, and differential equations are discretized by using the simultaneous collocation method. By this means, a MLDO problem is converted to a nonconvex mixed-integer nonlinear program (MINLP). The resulting nonconvex MINLP has a relatively small number of discrete variables and many highly nonlinear constraints, thus we use nonlinear branch and bound solvers to obtain good solutions. Two case study examples are considered, corresponding to a flowshop and a jobshop plant. We compare the integrated optimization approach with a base case approach, where the control recipes are predetermined and schedules are designed by solving mixed-integer linear programs (MILPs). In both cases, dynamic optimization of unit operations has enabled significant changes in production schedule. The optimized results strongly vouch for the integration method in process profit improvement.

In Chapters 4 and 5, dynamic reactor models have been developed for a class of semi-batch ring-opening polymerization reactors including polyether polyols. These rigorous models are based on fundamental first-principles such as mass and heat balances, polymerization reaction kinetics, phase equilibriums, etc. They enjoy high accuracies with respect to real plants. In the modeling phase, we have applied the null space projection method to deal with stiffness issues of equilibrium reaction systems with synergistic fast and slow reactions. The null space method introduces linear transformations to isolate fast equilibrium reactions, and models them as algebraic equations that describe their quasi-steady states. Also, reactor models based on the method of moments are developed besides the population balance models. Population balance models record the complete chain length distribution, which lead to large-scale models, especially for high molecular weight polymers. Moment models introduce principle moments of the distributions, and

track average polymer properties by moment balances that are relatively small-scale and independent of chain lengths. Validation of the developed reactor models is performed by comparing reactor pressure predictions with historical plant data. Satisfactory matches are obtained in all cases. Next, polymerization recipes are optimized to minimize polymerization time lengths by adjusting the reactor temperature and monomer feed policy over time. The model constraints in the recipe optimization problems consist of the reactor models, as well as equations for process safety and product quality requirements. For model solution, the reactor models are discretized by using the collocation method and the recipe optimization problems are solved as NLPs. Case studies are carried out for both homopolymerization and copolymerization, and considerable savings in polymerization times are achieved. In addition, we compare the population balance model with the moment model for copolymerization recipe optimization. The two models result in very close optimal solutions, but the moment model is more efficient in computation. Therefore, the moment model is preferred to be used in integrated problems.

In Chapter 6, we study scheduling formulations based on the discrete time resource task network (RTN). A number of extensions have been added to the conventional RTN model to enhance its modeling capability and solution efficiency, including multi-extent resource balance, resource limit balance, etc. Next, the extended RTN model is reformulated into the state space form via introducing lifted task history state variables. In a state space RTN model, process status is fully recorded by the task history states and process disturbances can be explicitly addressed in the task history evolution equation. The state space RTN model can be used for *on-line* scheduling applications, coupled with the rolling horizon scheme. In the case study, we show regular and reactive schedule designs with the state space RTN model. Also, the state space model has a coherent structure with lower level control models, which is advantageous in the context of integrated optimization problems.

Chapter 7 presents a discrete time optimization formulation for the integration of scheduling and dynamic optimization of general continuous/batch chemical processes. The method combines the state space RTN model with dynamic process unit models. The formulation gives a special structure that is exploited by the generalized Benders decomposition (GBD) method for efficient model solution. More specifically, the integrated formulation can be decomposed to a master scheduling problem and a primal dynamic optimization problem. The linking variables between the two sub-problems are the task history state variables. The integrated formulation is applied to a polymerization process including two parallel semi-batch reactors for ring-opening polymerization, continuous tanks and purification units. The two polymerization reactors share cooling utilities from the same source, and the price depends on the total consumption rate. Three different homopolymer products are made by the process. The optimization objective is to design the process schedule and reactor control profiles simultaneously to maximize the overall process profit. The moment reactor models are introduced to describe the polymerization dynamics. The case study results suggest improvements in plant profitability for the integrated approach, in contrast to the conventional approach, where polymerization recipes are individually optimized.

The publications resulting from this dissertation are listed as follows:

- Nie, Y., Biegler, L. T. and Wassick, J. M. (2012), Integrated scheduling and dynamic optimization of batch processes using state equipment networks. *AIChE J.*, 58: 3416-3432 [167].

- Nie, Y., Biegler, L. T., Villa, C. M. and Wassick, J. M. (2013), Reactor modeling and recipe optimization of polyether polyol processes: Polypropylene glycol. *AIChE J.*, 59: 2515-2529 [168].

- Nie, Y., Biegler, L. T., Villa, C. M. and Wassick, J. M. (2014), Reactor modeling and recipe optimization of ring-opening polymerization: Block copolymers. *Ind. Eng.*

*Chem. Res.*, 53 (18): 7434-7446 [169]

- Nie, Y., Biegler, L. T., Wassick, J. M. and Villa, C. M. (2014), Extended discrete-time resource task network formulation for the reactive scheduling of a mixed batch/continuous process. *Ind. Eng. Chem. Res.*, in press

## 8.2 Recommendations for Future Work

### 8.2.1 Polymerization Recipe Optimization

We have developed vapor-liquid equilibrium (VLE) equations based on the Flory-Huggins theory for polymerization reactors in Chapters 4 and 5, but this part of equations is not included in the recipe optimization problems. However, certain polymerization systems may have more specific constraints on the vapor phase composition or reactor pressure, thus require VLE equations to be added to the optimization formulation. Under this circumstance, the reactor venting event also needs to be modeled with equation-based methods. Introducing binary on/off variables is certainly able to meet this requirement, but it will significantly increase the model solution complexity. Another way for modeling this type of discontinuous elements in process systems is using complementary constraints. The problem remains continuous, once the complementary constraints are reformulated [170, 171].

Another interesting topic is to address uncertainties in recipe optimization. The optimal recipes obtained by deterministic optimization algorithms tend to drive the operating conditions very close to constraint boundaries to maximumly explore processing capabilities (for example, see the heat removal constraints in Figs. 4.3 on page 80 and 5.4 on page 107). However, these recipes may fail when considering the effect of process disturbances. A good strategy is to design a recipe that provides enough *back-offs* from the con-

straint boundaries, even if the *worst-case* disturbance occur. This research topic has been studied by Ma and Braatz [172], and Diehl et al. [173]. Sensitivity analysis techniques for differential-algebraic equations are used to estimate the effect of disturbances on process constraints. In addition, to further consider implementing the optimized recipes on-line, nonlinear model predictive control (NMPC) algorithms are required. A recent study by Mayne et al. has proposed a tube-based NMPC method [174]. The NMPC has a two-layer structure. The outer layer controller designs reference trajectories for the inner layer controller to track. This NMPC framework can be potentially coupled with the sensitivity analysis for disturbances, and therefore leads to a unified *on-line* recipe optimization approach.

### 8.2.2 Integrated Formulation

We have proposed integration formulations based on the continuous time and the discrete time representation in Chapter 3 and Chapter 7, respectively. Continuous time models have fewer discrete variables but additional timing constraints than discrete formulations. Optimizing task lengths is more straightforward with continuous time formulations, while discrete time models are easier to include details for process control. A hybrid grid [175] that take the complementary strength of the two time representations can be advantageous in the context of integration. For modeling process dynamics, reduced order models [176] are good alternatives to first-principles models, at least for the purpose of interfacing with scheduling equations. A possible dual-model structure can be employed in the integrated formulation, where scheduling decisions such as sequencing and timing are made with reduced order models and control policies are later refined by detailed first-principles models.

### 8.2.3 Decomposition Method

In Chapter 7 we apply the generalized Benders decomposition (GBD) methods to solve the integrated problem. It is also worth investigating the GBD method for the continuous time model in Chapter 3. Moreover, the Benders cuts are derived from solving discretized NLPs that have many degenerate constraints from the scheduling equations. The Lagrangian multipliers are in fact non-unique. It could be helpful to examine the effect of these non-unique multipliers on the tightness of the Benders cuts. Also, applying Benders cuts to non-convex problems takes the risk of cutting off feasible regions. Deriving cuts from relaxed convex NLPs can be a safer strategy to adopt.

### 8.2.4 On-line Implementation

The ultimate goal is to implement an on-line system that performs the integrated optimization in real time. So far, the computation bottleneck still remains a serious barrier. Progress has been made in nonlinear model predictive control algorithms that have enabled on-line implementation of dynamic optimization [177, 178]. Fast updates to control polices are realized by using NLP sensitivities [179]. However, methods for quick updates to production schedules are not currently available. A viable approach can be established by shifting the computational work from on-line to off-line. The off-line method derives rules for the on-line optimizer to adapt to disruptions occurred in schedule executions. Multi-parametric optimization approaches can be employed for this goal [180]. Also, necessary components for on-line optimizers such as state estimation and data reconciliation need to be considered.

# Bibliography


[1] J. Kallrath, "Planning and scheduling in the process industry," *OR Spectrum*, vol. 24, no. 3, pp. 219–250, 2002. 1.1

[2] C. A. Méndez, J. Cerdá, I. E. Grossmann, I. Harjunkoski, and M. Fahl, "State-of-the-art review of optimization methods for short-term scheduling of batch processes," *Computers & Chemical Engineering*, vol. 30, no. 6–7, pp. 913–946, 2006. 1.1, 1.2

[3] T. E. Marlin and A. N. Hrymak, "Real-time operations optimization of continuous processes," in *American Institute of Chemical Engineering Symposium Series - Fifth International Conference on Chemical Process Control*, vol. 93, 1997, pp. 85–112. 1.1, 1.2

[4] J. Kadam, W. Marquardt, M. Schlegel, T. Backx, O. Bosgra, P.-J. Brouwer, G. Dünnebier, D. van Hessem, A. Tiagounov, and S. De Wolf, "Towards integrated dynamic real-time optimization and control of industrial processes," in *Proceedings Foundations of Computer-Aided Process Operations (FOCAPO2003)*, 2003, pp. 593–596. 1.1

[5] S. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733 – 764, 2003. 1.1, 1.2

[6] J. Ziegler and N. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, pp. 759–768, 1942. 1.1

[7] I. E. Grossmann and A. W. Westerberg, "Research challenges in process systems

engineering," *AIChE Journal*, vol. 46, no. 9, pp. 1700–1703, 2000. 1.2

[8] J. F. Shapiro, "Challenges of strategic supply chain planning and modeling," *Computers & Chemical Engineering*, vol. 28, no. 6-7, pp. 855–861, 2004. 1.2

[9] I. E. Grossmann, "Enterprise-wide optimization: A new frontier in process systems engineering," *AIChE Journal*, vol. 51, no. 7, pp. 1846–1857, 2005. 1.2

[10] C. A. Floudas and X. Lin, "Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review," *Computers & Chemical Engineering*, vol. 28, no. 11, pp. 2109–2129, 2004. 1.2, 3.2.1

[11] I. Harjunkoski, C. T. Maravelias, P. Bongers, P. M. Castro, S. Engell, I. E. Grossmann, J. Hooker, C. Mndez, G. Sand, and J. Wassick, "Scope for industrial applications of production scheduling models and solution methods," *Computers & Chemical Engineering*, vol. 62, pp. 161–193, 2014. 1.2

[12] C. T. Maravelias and C. Sung, "Integration of production planning and scheduling: Overview, challenges and opportunities," *Computers & Chemical Engineering*, vol. 33, no. 12, pp. 1919 – 1930, 2009. 1.2

[13] P. M. Verderame, J. A. Elia, J. Li, and C. A. Floudas, "Planning and scheduling under uncertainty: A review across multiple sectors," *Industrial & Engineering Chemistry Research*, vol. 49, no. 9, pp. 3993–4017, 2010. 1.2

[14] B. Srinivasan, S. Palanki, and D. Bonvin, "Dynamic optimization of batch processes I. Characterization of the nominal solution," *Computers & Chemical Engineering*, vol. 27, no. 1, pp. 1–26, 2003. 1.2

[15] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki, "Dynamic optimization of batch processes II. Role of measurements in handling uncertainty," *Computers & Chemical Engineering*, vol. 27, no. 1, pp. 27–44, 2003. 1.2

[16] L. T. Biegler, "An overview of simultaneous strategies for dynamic optimization," *Chemical Engineering and Processing: Process Intensification*, vol. 46, no. 11, pp. 1043–1053, 2007. 1.2, 2.2.1

[17] S. Kameswaran and L. T. Biegler, "Simultaneous dynamic optimization strategies: Recent advances and challenges," *Computers & Chemical Engineering*, vol. 30, no. 10–12, pp. 1560–1575, 2006. 1.2

[18] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: Theory and practicea survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989. 1.2

[19] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999. 1.2

[20] J. B. Rawlings, "Tutorial overview of model predictive control," *Control Systems, IEEE*, vol. 20, no. 3, pp. 38–52, 2000. 1.2

[21] B. W. Bequette, "Nonlinear control of chemical processes: a review," *Industrial & Engineering Chemistry Research*, vol. 30, no. 7, pp. 1391–1413, 1991. 1.2

[22] S. A. van den Heever and I. E. Grossmann, "A strategy for the integration of production planning and reactive scheduling in the optimization of a hydrogen supply network," *Computers & Chemical Engineering*, vol. 27, no. 12, pp. 1813 – 1839, 2003. 1.3

[23] M. E. Dogan and I. E. Grossmann, "A decomposition method for the simultaneous planning and scheduling of single-stage continuous multiproduct plants," *Industrial & Engineering Chemistry Research*, vol. 45, no. 1, pp. 299–315, 2006. 1.3

[24] Z. Li and M. G. Ierapetritou, "Integrated production planning and scheduling using a decomposition framework," *Chemical Engineering Science*, vol. 64, no. 16, pp. 3585–3597, 2009. 1.3

[25] G. M. Kopanos, L. Puigjaner, and C. T. Maravelias, "Production planning and scheduling of parallel continuous processes with product families," *Industrial & Engineering Chemistry Research*, vol. 50, no. 3, pp. 1369–1378, 2011. 1.3

[26] Y. Chu, F. You, J. M. Wassick, and A. Agarwal, "Integrated planning and scheduling under production uncertainties: Bi-level model formulation and hybrid solution method," *Computers & Chemical Engineering*, 2014, in press. 1.3

[27] M. Diehl, H. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, 2002. 1.3

[28] J. V. Kadam, M. Schlegel, W. Marquardt, R. L. Tousain, D. H. van Hessem, J. van den Berg, and O. H. Bosgra, "A two-level strategy of integrated dynamic optimization and control of industrial processesa case study," in *European Symposium on Computer Aided Process Engineering-12 35th European Symposium of the Working Party on Computer Aided Process Engineering*. Computer Aided Chemical Engineering, vol. 10, pp. 511–516, 2002. 1.3

[29] V. Adetola and M. Guay, "Integration of real-time optimization and model predictive control," *Journal of Process Control*, vol. 20, no. 2, pp. 125–133, 2010. 1.3

[30] L. T. Biegler and V. M. Zavala, "Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization," *Computers & Chemical Engineering*, vol. 33, no. 3, pp. 575–582, 2009. 1.3

[31] S. Engell and I. Harjunkoski, "Optimal operation: Scheduling, advanced control and their integration," *Computers & Chemical Engineering*, vol. 47, no. 0, pp. 121–133, 2012. 1.4

[32] M. H. Bassett, P. Dave, F. J. Doyle III, G. K. Kudva, J. F. Pekny, G. V. Reklaitis,

S. Subrahmanyam, D. L. Miller, and M. G. Zentner, "Perspectives on model based integration of process operations," *Computers & Chemical Engineering*, vol. 20, no. 6-7, pp. 821–844, 1996, fifth International Symposium on Process Systems Engineering. 1.4

[33] D. E. Shobrys and D. C. White, "Planning, scheduling and control systems: why cannot they work together," *Computers & Chemical Engineering*, vol. 26, no. 2, pp. 149–160, 2002. 1.4

[34] I. Harjunkoski, R. Nyström, and A. Horch, "Integration of scheduling and control: Theory or practice?" *Computers & Chemical Engineering*, vol. 33, no. 12, pp. 1909–1918, 2009. 1.4

[35] R. J. Allgor and P. I. Barton, "Mixed-integer dynamic optimization I: problem formulation," *Computers & Chemical Engineering*, vol. 23, no. 45, pp. 567 – 584, 1999. 1.4, 2.3

[36] V. Bansal, V. Sakizlis, R. Ross, J. D. Perkins, and E. N. Pistikopoulos, "New algorithms for mixed-integer dynamic optimization," *Computers & Chemical Engineering*, vol. 27, no. 5, pp. 647–668, 2003. 1.4, 2.3, 7.2.1

[37] L. T. Biegler and I. E. Grossmann, "Retrospective on optimization," *Computers & Chemical Engineering*, vol. 28, no. 8, pp. 1169–1192, 2004. 1.4

[38] I. E. Grossmann and L. T. Biegler, "Part II. future perspective on optimization," *Computers & Chemical Engineering*, vol. 28, no. 8, pp. 1193–1218, 2004. 1.4

[39] R. Nyström, R. Franke, I. Harjunkoski, and A. Kroll, "Production campaign planning including grade transition sequencing and dynamic optimization," *Computers & Chemical Engineering*, vol. 29, no. 10, pp. 2163–2179, 2005. 1.4

[40] A. Flores-Tlacuahuac and I. E. Grossmann, "Simultaneous cyclic scheduling and

control of a multiproduct cstr," *Industrial & Engineering Chemistry Research*, vol. 45, no. 20, pp. 6698–6712, 2006. 1.4

[41] S. Terrazas-Moreno, A. Flores-Tlacuahuac, and I. E. Grossmann, "Simultaneous cyclic scheduling and optimal control of polymerization reactors," *AIChE Journal*, vol. 53, no. 9, pp. 2301–2315, 2007. 1.4, 2.3

[42] A. Prata, J. Oldenburg, A. Kroll, and W. Marquardt, "Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor," *Computers & Chemical Engineering*, vol. 32, no. 3, pp. 463–476, 2008. 1.4

[43] J. Busch, J. Oldenburg, M. Santos, A. Cruse, and W. Marquardt, "Dynamic predictive scheduling of operational strategies for continuous processes using mixed-logic dynamic optimization," *Computers & Chemical Engineering*, vol. 31, no. 56, pp. 574 – 587, 2007. 1.4

[44] Y. Chu and F. You, "Online integration of scheduling and control for cyclic production in CSTR," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec 2012, pp. 7007–7012. 1.4

[45] J. Zhuge and M. G. Ierapetritou, "Integration of scheduling and control with closed loop implementation," *Industrial & Engineering Chemistry Research*, vol. 51, no. 25, pp. 8550–8565, 2012. 1.4

[46] T. Bhatia and L. T. Biegler, "Dynamic optimization in the design and scheduling of multiproduct batch plants," *Industrial & Engineering Chemistry Research*, vol. 35, pp. 2234–2246, 1996. 1.4, 3.3

[47] B. V. Mishra, E. Mayer, J. Raisch, and A. Kienle, "Short-term scheduling of batch processes. A comparative study of different approaches," *Industrial & Engineering Chemistry Research*, vol. 44, no. 11, pp. 4022–4034, 2005. 1.4

[48] J. Romero, A. Espuña, F. Friedler, and L. Puigjaner, "A new framework for batch

process optimization using the flexible recipe," *Industrial & Engineering Chemistry Research*, vol. 42, no. 2, pp. 370–379, 2003. 1.4

[49] Y. Chu and F. You, "Integrated scheduling and dynamic optimization of sequential batch processes with online implementation," *AIChE Journal*, vol. 59, no. 7, pp. 2379–2406, 2013. 1.4

[50] ——, "Integrated scheduling and dynamic optimization of complex batch processes with general network structure using a generalized Benders decomposition approach," *Industrial & Engineering Chemistry Research*, vol. 52, no. 23, pp. 7867–7885, 2013. 1.4

[51] E. Capón-García, G. Guillén-Gosálbez, and A. Espuña, "Integrating process dynamics within batch process scheduling via mixed-integer dynamic optimization," *Chemical Engineering Science*, vol. 102, pp. 139–150, 2013. 1.4, 2.3

[52] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999. 1.4

[53] E. Gallestey, A. Stothert, D. Castagnoli, G. Ferrari-Trecate, and M. Morari, "Using model predictive control and hybrid systems for optimal scheduling of industrial processes," *Automatisierungstechnik*, vol. 51, no. 6, pp. 285–293, 2003. 1.4

[54] C. de Prada, I. E. Grossmann, D. Sarabia, and S. Cristea, "A strategy for predictive control of a mixed continuous batch process," *Journal of Process Control*, vol. 19, no. 1, pp. 123–137, 2009. 1.4

[55] M. L. Pinedo, *Scheduling: theory, algorithms, and systems*. Springer, 2012. 2.1

[56] C. T. Maravelias, "General framework and modeling approach classification for chemical production scheduling," *AIChE Journal*, vol. 58, no. 6, pp. 1812–1828, 2012. 2.1

[57] J. M. Pinto and I. E. Grossmann, "A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants," *Industrial & Engineering Chemistry Research*, vol. 34, no. 9, pp. 3037–3051, 1995. 2.1.1, 2.1.2

[58] C. T. Maravelias and I. E. Grossmann, "New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants," *Industrial & Engineering Chemistry Research*, vol. 42, no. 13, pp. 3056–3074, 2003. 2.1.1

[59] M. G. Ierapetritou and C. A. Floudas, "Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes," *Industrial & Engineering Chemistry Research*, vol. 37, no. 11, pp. 4341–4359, 1998. 2.1.1

[60] ——, "Effective continuous-time formulation for short-term scheduling. 2. Continuous and semicontinuous processes," *Industrial & Engineering Chemistry Research*, vol. 37, no. 11, pp. 4360–4374, 1998. 2.1.1

[61] C. A. Méndez, G. P. Henning, and J. Cerdá, "An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities," *Computers & Chemical Engineering*, vol. 25, no. 4, pp. 701–711, 2001. 2.1.1

[62] ——, "Optimal scheduling of batch plants satisfying multiple product orders with different due-dates," *Computers & Chemical Engineering*, vol. 24, no. 9, pp. 2223–2245, 2000. 2.1.1

[63] E. Kondili, C. C. Pantelides, and R. W.H. Sargent, "A general algorithm for short-term scheduling of batch operations–I. MILP formulation," *Computers & Chemical Engineering*, vol. 17, no. 2, pp. 211–227, 1993. 2.1.2

[64] C. C. Pantelides, "Unified frameworks for optimal process planning and schedul-

ing," in *Proceedings on the second conference on foundations of computer aided operations*. Cache Publications New York, 1994, pp. 253–274. 2.1.2, 6.3

[65] G. Schilling and C. C. Pantelides, "A simple continuous-time process scheduling formulation and a novel solution algorithm," *Computers & Chemical Engineering*, vol. 20, pp. 1221–1226, 1996. 2.1.2, 6.3

[66] T. Majozi and X. X. Zhu, "A novel continuous-time MILP formulation for multipurpose batch plants. 1. Short-term scheduling," *Industrial & Engineering Chemistry Research*, vol. 40, no. 25, pp. 5935–5949, 2001. 2.1.2

[67] E. M. Smith, "On the optimal design of continuous processes," Ph.D. dissertation, Imperial College of Science, Technology and Medicine, London, UK, 1996. 2.1.2, 3.2

[68] H. Yeomans and I. E. Grossmann, "A systematic modeling framework of superstructure optimization in process synthesis," *Computers & Chemical Engineering*, vol. 23, no. 6, pp. 709–731, 1999. 2.1.2

[69] R. Karp, "Reducibility among combinatorial problems," in *50 Years of Integer Programming 1958-2008*, M. Jnger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds. Springer Berlin Heidelberg, 2010, pp. 219–241. 2.1.3

[70] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*. Wiley New York, 1988. 2.1.3

[71] R. E. Gomory, "An algorithm for integer solutions to linear programs," *Recent advances in mathematical programming*, vol. 64, pp. 260–302, 1963. 2.1.3

[72] E. Balas, S. Ceria, and G. Cornuéjols, "A lift-and-project cutting plane algorithm for mixed 0–1 programs," *Mathematical programming*, vol. 58, no. 1-3, pp. 295–324, 1993. 2.1.3

[73] M. Padberg and G. Rinaldi, "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM review*, vol. 33, no. 1, pp. 60–100, 1991. 2.1.3

[74] CPLEX, *User's manual for CPLEX*, 12th ed., IBM ILOG, 2009. [Online]. Available: ftp://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmancplex.pdf 2.1.3

[75] Gurobi, *Gurobi optimizer reference manual*, Gurobi Optimization, 2012. [Online]. Available: http://www.gurobi.com 2.1.3

[76] L. T. Biegler, *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010. 2.2, 2.2.2

[77] V. S. Vassiliadis, R. W.H. Sargent, and C. C. Pantelides, "Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints," *Industrial & Engineering Chemistry Research*, vol. 33, no. 9, pp. 2111–2122, 1994. 2.2.1

[78] ——, "Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints," *Industrial & Engineering Chemistry Research*, vol. 33, no. 9, pp. 2123–2133, 1994. 2.2.1

[79] W. F. Feehery and P. I. Barton, "Dynamic optimization with state variable path constraints," *Computers & Chemical Engineering*, vol. 22, no. 9, pp. 1241–1256, 1998. 2.2.1

[80] M. Schlegel, K. Stockmann, T. Binder, and W. Marquardt, "Dynamic optimization using adaptive control vector parameterization," *Computers & Chemical Engineering*, vol. 29, no. 8, pp. 1731–1751, 2005. 2.2.1

[81] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of

optimal control problems," in *The 9th IFAC world congress*, Budapest, 1984. 2.2.1, 2.2.1

[82] J. E. Cuthrell and L. T. Biegler, "On the optimization of differential-algebraic process systems," *AIChE Journal*, vol. 33, no. 8, pp. 1257–1270, 1987. 2.2.1, 2.2.1

[83] H. G. Bock, M. Diehl, D. B. Leineweber, and J. P. Schlöder, "A direct multiple shooting method for real-time optimization of nonlinear dae processes," in *Nonlinear Model Predictive Control*, ser. Progress in Systems and Control Theory, F. Allgöwer and A. Zheng, Eds. Birkhäuser Basel, 2000, vol. 26, pp. 245–267. 2.2.1

[84] M. Caracotsios and W. E. Stewart, "Sensitivity analysis of initial value problems with mixed ODEs and algebraic equations," *Computers & Chemical Engineering*, vol. 9, no. 4, pp. 359–365, 1985. 2.2.1

[85] Y. Cao, S. Li, L. Petzold, and R. Serban, "Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution," *SIAM Journal on Scientific Computing*, vol. 24, no. 3, pp. 1076–1089, 2003. 2.2.1

[86] B. A. Finlayson, *The method of weighted residuals and variational principles: with application in fluid mechanics, heat and mass transfer*. New York, NY: Academic Press, 1972. 2.2.1

[87] U. M. Ascher, R. Mattheij, and R. D. Russell, *Numerical solution of boundary value problems for ordinary differential equations*. New Jersey, US: Prentice Hall, 1988. 2.2.1

[88] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004. 2.2.1

[89] J. E. Cuthrell and L. T. Biegler, "Simultaneous optimization and solution methods for batch reactor control profiles," *Computers & Chemical Engineering*, vol. 13, no. 12, pp. 49 – 62, 1989. 2.2.1

[90] A. Flores-Tlacuahuac, L. T. Biegler, and E. Saldívar-Guerra, "Dynamic optimization of HIPS open-loop unstable polymerization reactors," *Industrial & Engineering Chemistry Research*, vol. 44, no. 8, pp. 2659–2674, 2005. 2.2.1, 4.1

[91] V. M. Zavala, C. D. Laird, and L. T. Biegler, "Fast implementations and rigorous models: Can both be accommodated in NMPC?" *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 800–815, 2008. 2.2.1

[92] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed. New York, NY: Springer, 1999. 2.2.2

[93] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 979–1006, 2002. 2.2.2

[94] R. Fletcher and S. Leyffer, "Nonlinear programming without a penalty function," *Mathematical Programming*, vol. 91, no. 2, pp. 239–269, 2002. 2.2.2

[95] R. Fletcher, S. Leyffer, and P. L. Toint, "On the global convergence of a Filter-SQP algorithm," *SIAM Journal on Optimization*, vol. 13, no. 1, pp. 44–59, 2002. 2.2.2

[96] A. S. Drud, "CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems," *Mathematical Programming*, vol. 31, no. 2, pp. 153–191, 1985. 2.2.2

[97] ——, "CONOPT–a large-scale GRG code," *ORSA Journal on Computing*, vol. 6, no. 2, pp. 207–216, 1994. 2.2.2

[98] B. A. Murtagh and M. A. Saunders, "Large-scale linearly constrained optimization," *Mathematical Programming*, vol. 14, no. 1, pp. 41–72, 1978. 2.2.2

[99] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-

search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006. 2.2.2

[100] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999. 2.2.2

[101] R. J. Allgor and P. I. Barton, "Mixed-integer dynamic optimization," *Computers & Chemical Engineering*, vol. 21, Supplement, pp. S451 – S456, 1997, supplement to Computers and Chemical Engineering 6th International Symposium on Process Systems Engineering and 30th European Symposium on Computer Aided Process Engineering. 2.3

[102] P. I. Barton, R. J. Allgor, W. F. Feehery, and S. Galán, "Dynamic optimization in a discontinuous world," *Industrial & Engineering Chemistry Research*, vol. 37, no. 3, pp. 966–981, 1998. 2.3

[103] B. Chachuat, A. B. Singer, and P. I. Barton, "Global mixed-integer dynamic optimization," *AIChE Journal*, vol. 51, no. 8, pp. 2235–2253, 2005. 2.3

[104] ——, "Global methods for dynamic optimization and mixed-integer dynamic optimization," *Industrial & Engineering Chemistry Research*, vol. 45, no. 25, pp. 8373–8392, 2006. 2.3

[105] Y. Chu and F. You, "Integration of scheduling and control with online closed-loop implementation: Fast computational strategy and large-scale global optimization algorithm," *Computers & Chemical Engineering*, vol. 47, pp. 248–268, 2012. 2.3

[106] M. R. Bussieck and A. S. Drud, "SBB: A new solver for mixed integer nonlinear programming," in *http://www.gams.com/presentations/or01/sbb.pdf*, 2001. 2.3

[107] I. E. Grossmann, J. Viswanathan, A. Vecchietti, R. Raman, and E. Kalvelagen,

*GAMS/DICOPT: A discrete continuous optimization package*, GAMS Development Corporation, Washington DC, 2002. 2.3

[108] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, "An algorithmic framework for convex mixed integer nonlinear programs," *Discrete Optimization*, vol. 5, no. 2, pp. 186–204, 2008. 2.3

[109] T. Westerlund and K. Lundqvist, "Alpha-ECP, version 5.04: An interactive MINLP-solver based on the extended cutting plane method," Process Design Laboratory, Abo Akademi University, Tech. Rep., 2003. 2.3

[110] O. K. Gupta and A. Ravindran, "Branch and bound experiments in convex nonlinear integer programming," *Management Science*, pp. 1533–1546, 1985. 2.3

[111] A. M. Geoffrion, "Generalized Benders decomposition," *Journal of optimization theory and applications*, vol. 10, no. 4, pp. 237–260, 1972. 2.3

[112] M. A. Duran and I. E. Grossmann, "An outer-approximation algorithm for a class of mixed-integer nonlinear programs," *Mathematical Programming*, vol. 36, no. 3, pp. 307–339, 1986. 2.3

[113] T. Westerlund and F. Pettersson, "An extended cutting plane method for solving convex MINLP problems," *Computers & Chemical Engineering*, vol. 19, pp. 131–136, 1995. 2.3

[114] I. Quesada and I. E. Grossmann, "An LP/NLP based branch and bound algorithm for convex MINLP optimization problems," *Computers & Chemical Engineering*, vol. 16, no. 10-11, pp. 937–947, 1992. 2.3

[115] S. Terrazas-Moreno, A. Flores-Tlacuahuac, and I. E. Grossmann, "Lagrangean heuristic for the scheduling and control of polymerization reactors," *AIChE Journal*, vol. 54, no. 1, pp. 163–182, 2008. 2.3

[116] J. Oldenburg, W. Marquardt, D. Heinz, and D. B. Leineweber, "Mixed-logic dynamic optimization applied to batch distillation process design," *AIChE Journal*, vol. 49, no. 11, pp. 2900–2917, 2003. 2.3

[117] I. E. Grossmann, "Review of nonlinear mixed-integer and disjunctive programming techniques," *Optimization and Engineering*, vol. 3, no. 3, pp. 227–252, 2002. 2.3

[118] S. L. Janak, X. Lin, and C. A. Floudas, "Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: Resource constraints and mixed storage policies," *Industrial & Engineering Chemistry Research*, vol. 43, no. 10, pp. 2516–2533, 2004. 3.1

[119] E. M. Smith and C. C. Pantelides, "Design of reaction/separation networks using detailed models," *Computers & Chemical Engineering*, vol. 19, pp. 83–88, 1995. 3.2

[120] M. A. Shaik and C. A. Floudas, "Improved unit-specific event-based continuous-time model for short-term scheduling of continuous processes: Rigorous treatment of storage requirements," *Industrial & Engineering Chemistry Research*, vol. 46, no. 6, pp. 1764–1779, 2007. 3.2.1, 3.2.3

[121] S. Lee and I. E. Grossmann, "New algorithms for nonlinear generalized disjunctive programming," *Computers & Chemical Engineering*, vol. 24, no. 9, pp. 2125–2142, 2000. 3.2.2

[122] ——, "Logic-based modeling and solution of nonlinear discrete/continuous optimization problems," *Annals of Operations Research*, vol. 139, no. 1, pp. 267–288, 2005. 3.2.2

[123] J. Li and C. A. Floudas, "Optimal event point determination for short-term scheduling of multipurpose batch plants via unit-specific event-based continuous-time approaches," *Industrial & Engineering Chemistry Research*, vol. 49, no. 16, pp. 7446–7469, 2010. 3.2.3

[124] C.-W. Hui, "Optimizing chemical processes with discontinuous function – a novel formulation," *Computers & Chemical Engineering*, vol. 23, Supplement, pp. S479–S482, 1999, European Symposium on Computer Aided Process Engineering Proceedings of the European Symposium. 3.2.4

[125] A. Brooke, D. Kendrick, A. Meeraus, R. Raman, and R. Rosenthal, *GAMS a User's Guide*, GAMS Development Corporation, Washington DC, US, 2006. 3.3

[126] M. Ionescu, *Chemistry and technology of polyols for polyurethanes*. Shropshire, UK: Smithers Rapra Technologys, 2005. 4.1

[127] R. Herrington and K. Hock, *Dow Polyurethane Flexible Foams*. Midland, MI: Dow Chemical Co., 1997. 4.1, B.2

[128] R. M. Guibert, C. A. Plank, and E. R. Gerhard, "Kinetics of propylene oxide-oxypropylated glycerol reaction," *Industrial and Engineering Chemistry Process Design and Development*, vol. 10, no. 4, pp. 497–500, 1971. 4.1, B.2

[129] M. Di Serio, R. Tesser, A. Dimiccoli, and E. Santacesaria, "Kinetics of ethoxylation and propoxylation of ethylene glycol catalyzed by KOH," *Industrial & Engineering Chemistry Research*, vol. 41, no. 21, pp. 5196–5206, 2002. 4.1, B.2

[130] D. M. Simons and J. J. Verbanc, "The polymerization of propylene oxide," *Journal of Polymer Science*, vol. 44, no. 144, pp. 303–311, 1960. 4.1

[131] G.-E. Yu, A. J. Masters, F. Heatley, C. Booth, and T. G. Blease, "Anionic polymerisation of propylene oxide. investigation of double-bond and head-to-head content by nmr spectroscopy," *Macromolecular Chemistry and Physics*, vol. 195, no. 5, pp. 1517–1538, 1994. 4.1, 4.2.1

[132] G. Wegener, M. Brandt, L. Duda, J. Hofmann, B. Klesczewski, D. Koch, R.-J. Kumpf, H. Orzesek, H.-G. Pirkl, C. Six, C. Steinlein, and M. Weisbeck, "Trends in industrial

catalysis in the polyurethane industry," *Applied Catalysis A: General*, vol. 221, no. 1-2, pp. 303–335, 2001. 4.1

[133] M. Di Serio, R. Tesser, and E. Santacesaria, "Comparison of different reactor types used in the manufacture of ethoxylated, propoxylated products," *Industrial & Engineering Chemistry Research*, vol. 44, no. 25, pp. 9482–9489, 2005. 4.1

[134] V. M. Zavala and L. T. Biegler, "Optimization-based strategies for the operation of low-density polyethylene tubular reactors: nonlinear model predictive control," *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1735–1746, 2009. 4.1, 4.4.3

[135] C. Chatzidoukas, C. Kiparissides, J. D. Perkins, and E. N. Pistikopoulos, "Optimal grade transition campaign scheduling in a gas-phase polyolefin FBR using mixed integer dynamic optimization," *Computer Aided Chemical Engineering*, vol. 14, pp. 71–76, 2003. 4.1

[136] D. Bonvin, L. Bodizs, and B. Srinivasan, "Optimal grade transition for polyethylene reactors via NCO tracking," *Chemical Engineering Research and Design*, vol. 83, no. 6, pp. 692–697, 2005. 4.1

[137] V. M. Zavala, A. Flores-Tlacuahuac, and E. Vivaldo-Lima, "Dynamic optimization of a semi-batch reactor for polyurethane production," *Chemical Engineering Science*, vol. 60, no. 11, pp. 3061–3079, 2005. 4.1

[138] W. Lin, L. T. Biegler, and A. M. Jacobson, "Modeling and optimization of a seeded suspension polymerization process," *Chemical Engineering Science*, vol. 65, no. 15, pp. 4350–4362, 2010. 4.1

[139] G. Gee, W. C. Higginson, K. J. Taylor, and M. W. Trenholme, "The polymerization of epoxides. Part III. The polymerization of propylene oxide by sodium alkoxides," *Journal of the Chemical Society*, pp. 4298–4303, 1961. 4.2.1, B.2

[140] F. Heatley, Y. Luo, J. Ding, R. H. Mobbs, and C. Booth, "A carbon-13 nuclear mag-

netic resonance study of the triad sequence structure of block and statistical copolymers of ethylene oxide and propylene oxide," *Macromolecules*, vol. 21, no. 9, pp. 2713–2721, 1988. 4.2.1

[141] K.-D. Hungenberg, *Handbook of Polymer Reaction Engineering*. New York, NY: Wiley Online Library, 2008, ch. 7, pp. 323–359. 4.2.1, 5.1

[142] C. M. Villa, "Reactor modeling for polymerization processes," *Industrial & Engineering Chemistry Research*, vol. 46, no. 18, pp. 5815–5823, 2007. 4.2.2

[143] S. D. Gagnon, *Polyethers, Propylene Oxide Polymers. Kirk-Othmer Encyclopedia of Chemical Technology*. New York, NY: Wiley, 2000. 4.2.2, 4.2.2

[144] P. J. Flory, *Principles of polymer chemistry*. New York, NY: Cornell Univ Pr, 1953. 4.2.2

[145] A. Kumar and P. Daoutidis, "Nonlinear dynamics and control of process systems with recycle," *Journal of Process Control*, vol. 12, no. 4, pp. 475–484, 2002. 4.2.3

[146] O. Abel and W. Marquardt, "Scenario-integrated modeling and optimization of dynamic systems," *AIChE Journal*, vol. 46, no. 4, pp. 803–823, 2000. 4.3

[147] J. W. Verwijs, W. J. Papadopulos, J. W. Weston, R. J. Elwell, and C. M. Villa, "Continuous process and system of producing polyether polyols," US Patent US20 080 221 281 A1, Mar. 25, 2008, US Patent App. 12/054,782. 4.3

[148] C. C. Pantelides and J. G. Renfro, "The online use of first-principles models in process operations: review, current status and future needs," *Computers & Chemical Engineering*, vol. 51, pp. 136–148, 2012. 4.4.1

[149] G. Odian, *Principles of Polymerization*. New York, NY: John Wiley and Sons, 1991. 5.1

[150] O. Abel, A. Helbig, W. Marquardt, H. Zwick, and T. Daszkowski, "Productivity

optimization of an industrial semi-batch polymerization reactor under safety constraints," *Journal of Process Control*, vol. 10, no. 4, pp. 351–362, 2000. 5.1

[151] W. H. Ray, "On the mathematical modeling of polymerization reactors," *Journal of Macromolecular Science, Part C: Polymer Reviews*, vol. 8, no. 1, pp. 1–56, 1972. 5.1

[152] E. Favre, Q. T. Nguyen, R. Clement, and J. Neel, "Application of flory-huggins theory to ternary polymer-solvents equilibria: A case study," *European polymer journal*, vol. 32, no. 3, pp. 303–309, 1996. 5.2.1

[153] F. R. Mayo and F. M. Lewis, "Copolymerization. I. A basis for comparing the behavior of monomers in copolymerization; the copolymerization of styrene and methyl methacrylate," *Journal of the American Chemical Society*, vol. 66, no. 9, pp. 1594–1601, 1944. 5.3

[154] J. M. Wassick, "Enterprise-wide optimization in an integrated chemical complex," *Computers & Chemical Engineering*, vol. 33, no. 12, pp. 1950–1963, 2009. 6.1

[155] J. M. Wassick and J. Ferrio, "Extending the resource task network for industrial applications," *Computers & Chemical Engineering*, vol. 35, no. 10, pp. 2124–2140, 2011. 6.1, 6.3.2

[156] J. M. Wassick, A. Agarwal, N. Akiya, J. Ferrio, S. Bury, and F. You, "Addressing the operational challenges in the development, manufacture, and supply of advanced materials and performance products," *Computers & Chemical Engineering*, vol. 47, pp. 157–169, 2012. 6.1

[157] K. L. Yee and N. Shah, "Improving the efficiency of discrete time scheduling formulation," *Computers & Chemical Engineering*, vol. 22, pp. 403–410, 1998. 6.1

[158] S. Velez and C. T. Maravelias, "Reformulations and branching methods for mixed-integer programming chemical production scheduling models," *Industrial & Engineering Chemistry Research*, vol. 52, no. 10, pp. 3832–3841, 2013. 6.1

[159] ——, "A branch-and-bound algorithm for the solution of chemical production scheduling MIP models using parallel computing," *Computers & Chemical Engineering*, vol. 55, pp. 28–39, 2013. 6.1

[160] S. Ferrer-Nadal, C. A. Méndez, M. Graells, and L. Puigjaner, "Optimal reactive scheduling of manufacturing plants with flexible batch recipes," *Industrial & Engineering Chemistry Research*, vol. 46, no. 19, pp. 6273–6283, 2007. 6.1

[161] Z. Li and M. G. Ierapetritou, "Reactive scheduling using parametric programming," *AIChE Journal*, vol. 54, no. 10, pp. 2610–2623, 2008. 6.1

[162] K. Subramanian, C. T. Maravelias, and J. B. Rawlings, "A state-space model for chemical production scheduling," *Computers & Chemical Engineering*, vol. 47, pp. 97–110, 2012. 6.1, 6.3.3, 6.3.3

[163] C. T. Maravelias and I. E. Grossmann, "Minimization of the makespan with a discrete-time state-task network formulation," *Industrial & Engineering Chemistry Research*, vol. 42, no. 24, pp. 6252–6257, 2003. 6.3.1

[164] P. M. Castro, I. Harjunkoski, and I. E. Grossmann, "Optimal short-term scheduling of large-scale multistage batch plants," *Industrial and Engineering Chemistry*, vol. 48, no. 24, pp. 11 002–11 016, 2009. 6.4.1

[165] N. V. Sahinidis and I. E. Grossmann, "Convergence properties of generalized benders decomposition," *Computers & Chemical Engineering*, vol. 15, no. 7, pp. 481 – 491, 1991. 7.2.3

[166] S. Velez and C. T. Maravelias, "Mixed-integer programming model and tightening methods for scheduling in general chemical production environments," *Industrial & Engineering Chemistry Research*, vol. 52, no. 9, pp. 3407–3423, 2013. 7.3

[167] Y. Nie, L. T. Biegler, and J. M. Wassick, "Integrated scheduling and dynamic op-

timization of batch processes using state equipment networks," *AIChE Journal*, vol. 58, no. 11, pp. 3416–3432, 2012. 8.1

[168] Y. Nie, L. T. Biegler, C. M. Villa, and J. M. Wassick, "Reactor modeling and recipe optimization of polyether polyol processes: Polypropylene glycol," *AIChE Journal*, vol. 59, pp. 2515–2529, 2013. 8.1

[169] ——, "Reactor modeling and recipe optimization of ring-opening polymerization: Block copolymers," *Industrial & Engineering Chemistry Research*, vol. 53, no. 18, pp. 7434–7446, 2014. 8.1

[170] A. U. Raghunathan and L. T. Biegler, "Mathematical programs with equilibrium constraints (MPECs) in process engineering," *Computers & Chemical Engineering*, vol. 27, no. 10, pp. 1381 – 1392, 2003. 8.2.1

[171] B. T. Baumrucker, J. G. Renfro, and L. T. Biegler, "MPEC problem formulations and solution strategies with chemical engineering applications," *Computers & Chemical Engineering*, vol. 32, no. 12, pp. 2903 – 2913, 2008. 8.2.1

[172] D. L. Ma and R. D. Braatz, "Worst-case analysis of finite-time control policies," *Control Systems Technology, IEEE Transactions on*, vol. 9, no. 5, pp. 766–774, 2001. 8.2.1

[173] M. Diehl, H. Bock, and E. Kostina, "An approximation technique for robust nonlinear optimization," *Mathematical Programming*, vol. 107, no. 1, pp. 213–230, 2006. 8.2.1

[174] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, "Tube-based robust nonlinear model predictive control," *International Journal of Robust and Nonlinear Control*, 2011. 8.2.1

[175] C. T. Maravelias, "Mixed-time representation for state-task network models," *Industrial & Engineering Chemistry Research*, vol. 44, no. 24, pp. 9129–9145, 2005. 8.2.2

[176] L. T. Biegler, Y.-d. Lang, and W. Lin, "Multi-scale optimization for process systems engineering," *Computers & Chemical Engineering*, vol. 60, pp. 17–30, 2014. 8.2.2

[177] V. M. Zavala and L. T. Biegler, "The advanced-step NMPC controller: Optimality, stability and robustness," *Automatica*, vol. 45, no. 1, pp. 86–93, 2009. 8.2.4

[178] L. Würth, R. Hannemann, and W. Marquardt, "Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization," *Journal of Process Control*, vol. 19, no. 8, pp. 1277–1288, 2009. 8.2.4

[179] A. V. Fiacco, "Sensitivity analysis for nonlinear programming using penalty methods," *Mathematical Programming*, vol. 10, no. 1, pp. 287–311, 1976. 8.2.4

[180] V. Dua and E. N. Pistikopoulos, "An algorithm for the solution of multiparametric mixed integer linear programming problems," *Annals of operations research*, vol. 99, no. 1, pp. 123–139, 2000. 8.2.4

[181] S. M. Bajorek and J. Schnelle, "Identification and experimental database for binary and multicomponent mixtures with potential for increasing overall cycle efficiency," Kansas State University, Tech. Rep., 2002. B.2

[182] C. L. Yaws. (2003) Yaws' handbook of thermodynamic and physical properties of chemical compounds. Knovel. New York, NY. [Online]. Available: http://www.knovel.com B.2

[183] R. Beaumont, B. Clegg, G. Gee, J. Herbert, D. Marks, R. Roberts, and D. Sims, "Heat capacities of propylene oxide and of some polymers of ethylene and propylene oxides," *Polymer*, vol. 7, no. 8, pp. 401–417, 1966. B.2

[184] C. L. Yaws, P. K. Narasimhan, and C. Gabbula. (2009) Yaws' handbook of antoine coefficients for vapor pressure (2nd electronic edition). Knovel. New York, NY. [Online]. Available: http://www.knovel.com B.2

[185] D. R. Stull, "Vapor pressure of pure substances. organic and inorganic compounds," *Industrial and Engineering Chemistry*, vol. 39, no. 4, pp. 517–540, 1947. B.2

# Appendices

# Appendix A

# Supplementary Information for Chapter 3

## A.1 The Flowshop Plant Example

### A.1.1 Models of Operations

#### Reaction

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

Material state transition:

$$FeedA \longrightarrow IntABC$$

Reaction formula:

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

DAE system:

$$\begin{aligned} \frac{dc_A}{dt} &= -uc_A^2 & c_A(0) &= 1 \\ \frac{dc_B}{dt} &= uc_A^2 - \beta u^\alpha c_B & c_B(0) &= 0 \\ \frac{dc_C}{dt} &= \beta u^\alpha c_B & c_c(0) &= 0 \end{aligned} \tag{A.1.1}$$

where scaled temperature $u$ is defined to substitute rate constants $k_1$, and $k_2$, and the unit used here is $m^3/(kg \cdot hr)$. While there is no energy balance equation, the differences

in activation energy can largely affect the trends of the operating temperature as we can observe from the case study results. The costs of reactions come from the usage of heating utility of unit price $p$. The cost function of reactors can be described as Eq. (A.1.2).

$$F = pb \int_0^{Tp} u(t)dt \tag{A.1.2}$$

Important parameters are tabulated as follows

| Reaction | $p$ $(MU/kg)$ | $\alpha$ | $\beta$ | $Tp^{min}$ | $Tp^{max}$ $(hr)$ | $Tp^{recipe}$ | $u^{min}$ | $u^{max}$ $m^3/(kg{\cdot}hr)$ | $u^{recipe}$ |
|---|---|---|---|---|---|---|---|---|---|
| | 0.3 | 2 | $3.875 \times 10^{-3}$ | 1.5 | 3 | 2 | 1.8 | 7.9 | 5 |

**Distillation**

Material state transitions:

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

The dynamic models of the distillation states are based on a simplified tray-type binary distillation column model (4 trays in our study), which has been set up by making following assumptions:

- ⋆ Binary distillation
- ⋆ Negligible tray holdup
- ⋆ Tray equilibrium with constant relative volatility $\alpha$
- ⋆ Equimolar overflow (constant vapor rate $V$)

Differential equations:

$$\begin{aligned} \dot{S} &= L - V & S(0) &= b \\ \dot{x_b} &= \frac{V(x_b - x_d)}{(R+1)S} & x_b(0) &= \eta \end{aligned} \tag{A.1.3}$$

Constant vapor flow:

$$V = kb \tag{A.1.4}$$

Composition balance for each tray:

$$\begin{aligned} &Lx_i + Vy_i = Lx_{i+1} + Vy_{i-1} \qquad \forall i = 2 \dots NT-1 \\ &Lx_N + Vy_N = Lx_d + Vy_{N-1} \\ &Lx_1 + Vy_1 = Lx_2 + Vy_b \end{aligned} \tag{A.1.5}$$

Phase equilibrium:

$$\begin{aligned} y_i &= \frac{\alpha x_i}{1+(\alpha-1)x_i} \qquad \forall i = 1 \dots NT \\ y_b &= \frac{\alpha x_b}{1+(\alpha-1)x_b} \end{aligned} \tag{A.1.6}$$

At the top of the column:

$$\begin{aligned} R &= \frac{L}{D} \\ L + D &= V \end{aligned} \tag{A.1.7}$$

Average purity:

$$\bar{x_d} \int_0^t D dt = S_0 x_0 - S(t) x_b(t) \tag{A.1.8}$$

Requirements for manufacturing:

$$\begin{aligned} R^{min} \leqslant R \leqslant R^{max} \\ \bar{x_d} \geqslant \bar{x_d}^{min} \end{aligned} \tag{A.1.9}$$

The cost function is defined as below:

$$F = pVT_p \tag{A.1.10}$$

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

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

**Filtration**

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

Material state transitions:

$$IntABC \longrightarrow IntAB + WasteC$$

Production and consumption:

$$\begin{aligned} R^p_{IntAB} &= (1 - \eta_{IntABC,C})b \\ R^p_{WstC} &= \eta_{IntABC,C}b \\ R^c_{IntABC} &= b \end{aligned} \tag{A.1.11}$$

Operating cost:

$$F = R^p_{IntAB} + 4R^p_{WstC} \tag{A.1.12}$$

**Reduced MILP Formulation**

The recipe-based models we use to compare with the integrated formulation are linear without dynamic and quality variables and constraints. The reduced problem can be reformulated as an MILP and solved for the same objective function as the integrated formulation. Here, instead of the original disjunctive DAEs, unit operations are described through Eq. (A.1.13).

$$\cdots \begin{bmatrix} w_{j,s,n} = 1 \\ R^p_{j,r,n} = \mu^p_{j,s} b_{j,n} \quad (\forall r \in \mathcal{R}^p_s) \\ R^c_{j,r,n} = \mu^c_{j,s} b_{j,n} \quad (\forall r \in \mathcal{R}^c_s) \\ R^p_{j,r,n} = 0 \quad (\forall r \notin \mathcal{R}^p_s) \\ R^c_{j,r,n} = 0 \quad (\forall r \notin \mathcal{R}^c_s) \\ Tp_{j,n} = \alpha_{j,s} + \beta_{j,s} b_{j,n} \\ F_{j,n} = \gamma_{j,s} b_{j,n} \end{bmatrix}_{s \in \mathcal{S}_j} \cdots \bigvee \begin{bmatrix} \sum_{s \in \mathcal{S}_j} w_{j,s,n} = 0 \\ R^p_{j,r,n} = 0 \\ R^c_{j,r,n} = 0 \\ Tp_{j,n} = 0 \\ F_{j,n} = 0 \end{bmatrix} \qquad \forall j \in \mathcal{J}, n \in \mathcal{N} \tag{A.1.13}$$

The parameters $(\alpha, \beta, \gamma, \mu^p, \mu^c)$ are obtained via simulation based on predefined recipes

which in this study we determine manually. For each sequence of operations to make a product, we start with simulating the first stage with known controls, processing times and initial conditions according to the recipe of the operation. Then the procedure is repeated for the subsequent stages, using the information from the recipe and the calculated precedent stage. Parameters for this example are listed as follows.

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

### A.1.2 Scheduling Parameters

Material information:

| Material | Component | | | Parameter | | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | $\overline{\eta}$ | $P_{(MU/kg)}$ | $E_{0(kg)}$ | $E^{max}{}_{(kg)}$ | $E^{min}{}_{(kg)}$ |
| $FeedA$ | ✓ | | | | 5 | $var$ | 400 | 0 |
| $IntABC$ | ✓ | ✓ | ✓ | | 0 | 0 | 400 | 0 |
| $IntAB$ | ✓ | ✓ | | | 0 | 0 | 400 | 0 |
| $WstC$ | | | ✓ | | 0 | 0 | 400 | 0 |
| $Prod1$ | ✓ | ✓ | | $B \geqslant 99.5\%$ | 30 | 0 | 400 | 0 |
| $Prod2$ | ✓ | ✓ | | $B \geqslant 99.7\%$ | 45 | 0 | 400 | 0 |
| $Rcy1$ | ✓ | ✓ | | | 0 | 0 | 400 | 0 |
| $Rcy2$ | ✓ | ✓ | | | 0 | 0 | 400 | 0 |

Equipment information:

| Unit | Operating states | $B^{max}{}_{(kg)}$ | $B^{min}{}_{(kg)}$ |
|---|---|---|---|
| $Reactor$ | $Rct$ | 60 | 30 |
| $Filter$ | $Fil$ | 60 | 30 |
| $Column$ | $Dis1, Dis2$ | 60 | 30 |

## A.2 The Jobshop Plant Example

### A.2.1 Models of Operations

**Reaction**

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

Material state transition:

$$HotA \longrightarrow IntADE$$

Reaction formula:

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

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

DAE system:

$$\begin{aligned} \frac{dc_A}{dt} &= -(u + \beta u^{\alpha})c_A \quad c_A(0) = 1 \\ \frac{dc_D}{dt} &= uc_A \qquad\qquad\quad c_D(0) = 0 \\ \frac{dc_E}{dt} &= \beta u^{\alpha} c_A \qquad\quad c_E(0) = 0 \end{aligned} \tag{A.2.1}$$

Important process data:

| Reaction 2 | $p$ $(MU/kg)$ | $\alpha$ | $\beta$ | $Tp^{min}$ | $Tp^{max}$ $(hr)$ | $Tp^{recipe}$ | $u^{min}$ | $u^{max}$ $(/hr)$ | $u^{recipe}$ |
|---|---|---|---|---|---|---|---|---|---|
| | 3 | 2 | 0.5 | 1.5 | 3 | 2 | 0.3 | 1.5 | 0.7 |

**Distillation**

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

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

**Filtration**

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

Material state transition:

$$IntADE \longrightarrow IntDE + WasteA$$

Production and consumption:

$$\begin{aligned} R^p_{IntDE} &= (1 - \eta_{IntADE,A})b \\ R^p_{WstA} &= \eta_{IntADE,A} b \\ R^c_{IntADE} &= b; \end{aligned} \tag{A.2.2}$$

Operating cost:

$$F = R^p_{IntDE} + 4R^p_{WstA} \tag{A.2.3}$$

### Heating and Packaging

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

### Reduced MILP Formulation

Unit operations are described through Eq. (A.1.13), where the parameters are listed as follows.

| | $Hting$ | $Rct1$ | $Rct2$ | $Fil1$ | $Fil2$ | $Dis1$ | $Dis2$ | $Pck1$ | $Pck2$ | $Pck3$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $\alpha_{(hr)}$ | | | | | | | | | | |
| $Heater$ | 1 | | | | | | | | | |
| $Reactor1$ | | 2 | 2 | | | | | | | |
| $Reactor2$ | | 2 | 2 | | | | | | | |
| $Filter1$ | | | | 1 | | | | | | |
| $Filter2$ | | | | | 1 | | | | | |
| $Column$ | | | | | | 2 | 1.5 | | | |
| $Line1$ | | | | | | | | 0.667 | 0.667 | 0.667 |
| $Line2$ | | | | | | | | 0.667 | 0.667 | 0.667 |
| $\beta_{(hr/kg)}$ | | | | | | | | | | |
| $Heater$ | 0.0125 | | | | | | | | | |
| $Reactor1$ | | 0 | 0 | | | | | | | |
| $Reactor2$ | | 0 | 0 | | | | | | | |
| $Filter1$ | | | | 0.0125 | | | | | | |
| $Filter2$ | | | | | 0.0125 | | | | | |
| $Column$ | | | | | | 0 | 0 | | | |
| $Line1$ | | | | | | | | 0.0167 | 0.0167 | 0.0167 |
| $Line2$ | | | | | | | | 0.0223 | 0.0223 | 0.0223 |
| $\gamma_{(MU/kg)}$ | | | | | | | | | | |
| $Heater$ | 1 | | | | | | | | | |
| $Reactor1$ | | 3 | 4.2 | | | | | | | |
| $Reactor2$ | | 3 | 4.2 | | | | | | | |
| $Filter1$ | | | | 1.405 | | | | | | |
| $Filter2$ | | | | | 1.453 | | | | | |
| $Column$ | | | | | | 4.938 | 3.704 | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $Line1$ | | | | | | | 1 | 1 | 1 |
| $Line2$ | | | | | | | 0.9 | 0.9 | 0.9 |
| $\mu^{p/c}{}_{(+/-)}$ | | | | | | | | | |
| $FeedA$ | $-1$ | $-1$ | | | | | | | |
| $HotA$ | 1 | | $-1$ | | | | | | |
| $IntABC$ | | 1 | | $-1$ | | | | | |
| $IntADE$ | | | 1 | | $-1$ | | | | |
| $IntAB$ | | | | 0.865 | | $-1$ | | | |
| $IntDE$ | | | | | 0.849 | | $-1$ | | $-1$ |
| $WstC$ | | | | 0.135 | | | | | |
| $WstA$ | | | | | 0.151 | | | | |
| $RProd1$ | | | | | | 0.658 | | $-1$ | |
| $RProd2$ | | | | | | | 0.617 | | $-1$ |
| $Rcy1$ | | | | | | 0.342 | | | |
| $Rcy2$ | | | | | | | 0.383 | | |
| $Prod1$ | | | | | | | | 1 | |
| $Prod2$ | | | | | | | | | 1 |
| $Prod3$ | | | | | | | | | |

### A.2.2 Scheduling Parameters

Material information:

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

Equipment information:

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

# Appendix B

# Supplementary Information for Chapter 4

## B.1 Reformulation of the Propoxylation Model

The nullspace projection method is applied to reformulate the propoxylation model. First, the population balance equations involving the exchange reactions are converted into a matrix representation, and then the reformulation method can be systematically carried out.

### B.1.1 Model Matrix Representation

For each adduct set, the population is defined as the product of its concentration and the liquid volume, i.e.,

$$\mathrm{x}_n = [\mathrm{x_n}]V, \qquad x = \{\mathrm{G}, \mathrm{D}, \mathrm{Q}, \mathrm{R}\}, n = 0, 1, \ldots, N. \tag{B.1.1}$$

Similar definitions can be introduced for the monomer ($\mathrm{M}$) and water ($\mathrm{W}$). According to the form in Eq. (4.11), the state vector can be defined as

$$\mathrm{x}^{\mathrm{T}} = \begin{bmatrix} \mathrm{W} & \mathrm{M} & \mathrm{G}^{\mathrm{T}} & \mathrm{D}^{\mathrm{T}} & \mathrm{Q}^{\mathrm{T}} & \mathrm{R}^{\mathrm{T}} \end{bmatrix}. \tag{B.1.2}$$

Here, $\mathrm{G}$ represents the vector of population $\begin{bmatrix} \mathrm{G}_0 & \mathrm{G}_1 & \cdots & \mathrm{G}_N \end{bmatrix}^{\mathrm{T}}$, and similarly for the other polymeric species. Meanwhile, the reaction rate vector $r(\mathrm{x})$ consists of eight segments with regard to the hydrolysis reaction, as well as the initiation, propagation, transfer and exchange reactions of both the normal and unsat chains. Mathematically, it reads:

$$r^{\mathrm{T}} = \begin{bmatrix} r_1^{\mathrm{T}} & r_2^{\mathrm{T}} & \cdots & r_8^{\mathrm{T}} \end{bmatrix}; \tag{B.1.3a}$$

$$r_1 = \begin{bmatrix} V^{-1} k_h \mathrm{WM} \end{bmatrix}, \tag{B.1.3b}$$

$$r_2 = \begin{bmatrix} V^{-1} k_i \mathrm{G}_0 \mathrm{M} \end{bmatrix}, \tag{B.1.3c}$$

$$r_3 = \begin{bmatrix} V^{-1} k_p \mathrm{G}_1 \mathrm{M} & \cdots & V^{-1} k_p \mathrm{G}_{N-1} \mathrm{M} \end{bmatrix}^{\mathrm{T}}, \tag{B.1.3d}$$

$$r_4 = \begin{bmatrix} V^{-1} k_i \mathrm{Q}_0 \mathrm{M} \end{bmatrix}, \tag{B.1.3e}$$

$$r_5 = \begin{bmatrix} V^{-1} k_p \mathrm{Q}_1 \mathrm{M} & \cdots & V^{-1} k_p \mathrm{Q}_{N-1} \mathrm{M} \end{bmatrix}^{\mathrm{T}}, \tag{B.1.3f}$$

$$r_6 = \begin{bmatrix} V^{-1} k_t \mathrm{G}_0 \mathrm{M} & \cdots & V^{-1} k_t \mathrm{G}_N \mathrm{M} \end{bmatrix}^{\mathrm{T}}, \tag{B.1.3g}$$

$$r_7 = \begin{bmatrix} V^{-1} k_t \mathrm{Q}_0 \mathrm{M} & \cdots & V^{-1} k_t \mathrm{Q}_N \mathrm{M} \end{bmatrix}^{\mathrm{T}}, \tag{B.1.3h}$$

$$r_8 = \begin{bmatrix} V^{-1} k_e \mathrm{G}_n \mathrm{D}_m & V^{-1} k_e \mathrm{G}_n \mathrm{R}_m & V^{-1} k_e \mathrm{Q}_n \mathrm{D}_m & V^{-1} k_e \mathrm{Q}_n \mathrm{R}_m \end{bmatrix}^{\mathrm{T}}, \qquad n, m = 0, 1, \ldots, N. \tag{B.1.3i}$$

The equation system can be written in a comprehensive manner as shown below:

$$\frac{\mathrm{d}}{\mathrm{d}t} \left[\begin{array}{c} \mathrm{W} \\ \mathrm{M} \\ \hline \mathrm{G} \\ \hline \mathrm{D} \\ \hline \mathrm{Q} \\ \hline \mathrm{R} \end{array}\right] = \left[\begin{array}{c|cc} A_{11} & & \\ \hline & A_{21} & \\ A_{12} & -A_{21} & \\ & A_{22} & \\ & -A_{22} & \end{array}\right] \left[\begin{array}{c} r_1 \\ r_2 \\ \vdots \\ r_7 \\ \hline r_8 \end{array}\right] + \left[\begin{array}{c} \\ B \\ \\ \\ \\ \\ \\ \end{array}\right] F \tag{B.1.4}$$

In the partitioned coefficient matrix, $A_{11}$ corresponds to the reaction rates for water and monomers. For the polymers, $A_{12}$ represents the non-equilibrium reaction coefficients, from $r_1$ to $r_7$; $A_{21}$ includes the terms from the equilibrium (exchange) reactions for $\mathrm{G}$, and the sub-matrix for $\mathrm{D}$ is $-A_{21}$ since the population of $\mathrm{D}$ changes reversely; similarly, $A_{22}$ and $-A_{22}$ are defined for $\mathrm{Q}$ and $\mathrm{R}$ respectively. The input matrix $B = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \end{bmatrix}^{\mathrm{T}}$ as the feed contains only the monomer.

## B.1.2 Reformulated Model

The particular structure of Eq. (B.1.4) allows us to define the range $\mathcal{Y}$ and the null space matrix $\mathcal{Z}$ as:

$$\begin{bmatrix} \mathcal{Y} \mid \mathcal{Z} \end{bmatrix} = \left[\begin{array}{ccc|cc} I_2 & & & & \\ & I_{N+1} & & I_{N+1} & \\ & -I_{N+1} & & I_{N+1} & \\ & & I_{N+1} & & I_{N+1} \\ & & -I_{N+1} & & I_{N+1} \end{array}\right]. \tag{B.1.5}$$

| | A | B | C | D | Reference |
|---|---|---|---|---|---|
| Feed ($f$) | 0.92 | $8.87 \times 10^{-3}$ | $-3.10 \times 10^{-5}$ | $4.78 \times 10^{-8}$ | Yaws [182] |
| Bulk ($b$) | 1.10 | $2.72 \times 10^{-3}$ | 0 | 0 | Beaumont et al. [183] |

$$c_{p\,i} = A_i + B_i T + C_i T^2 + D_i T^3, \quad i = \{f, b\}$$

Table B.1: Heat capacity coefficients

It can be verified that $\mathcal{Z}^{\mathrm{T}} A_2 = 0$, and the reformulated system is obtained as stated in Eqs (4.16)- (4.18).

## B.2 Model Parameters

Heat capacities for enthalpy calculations are listed in Tab. B.1. The Antoine equation coefficients for vapor pressure calculations are tabulated in Tab. B.2. Other important parameters used are listed as below:

Universal gas constant $R = 8.314\ \mathrm{J/mol \cdot K}$;
Initial amount of nitrogen $n_{\mathrm{N_2}}$;
Maximum reactor pressure $P^{max}$;
Total reactor volume $V + \bar{V}$;
Interaction parameter $\chi$ (in the Flory-Huggins theory);
Liquid phase activity of water $a_{\mathrm{H_2O}}$;
Liquid phase activity of propylene glycol $a_{\mathrm{PG}}$.

The initial amount of nitrogen charged in the reactor, the maximum pressure and the reactor volume are given by the associated process specification. The interaction parameter is decided on our experience with the alkoxylation system. The liquid phase activities of water and propylene glycol are obtained from the technical report [181]. These parameters are adjusted to match the pressure profile from plant data. Moreover, in Tab. B.3, the kinetic constants are expressed with the Arrhenius temperature dependence $k_r = A_r exp(-\frac{E_r}{RT}), r = \{h, i, p, t\}$. The parameter values are obtained from published literature as noted in the last column, and some of them are adjusted to fit the plant pressure profile in our case study.

| | A | B | C | Reference |
|---|---|---|---|---|
| Water | 7.18 | 1723.64 | -40.07 | Yaws et al. [184] |
| PO | 6.28 | 1158.00 | -36.93 | Yaws et al. [184] |
| PG | 8.08 | 2692.19 | -14.97 | Stull [185] |

Table B.2: Antoine equation coefficients

| Model parameter | Unit | Reference |
|---|---|---|
| $A_h = 240420$ | $\mathrm{m}^3/\mathrm{mol} \cdot \mathrm{s}$ | Di Serio et al. [129] |
| $E_h = 82425$ | $\mathrm{J}/\mathrm{mol}$ | Di Serio et al. [129] |
| $A_i = 396400$ | $\mathrm{m}^3/\mathrm{mol} \cdot \mathrm{s}$ | Di Serio et al. [129] |
| $E_i = 77822$ | $\mathrm{J}/\mathrm{mol}$ | Di Serio et al. [129] |
| $A_p = 8504$ | $\mathrm{m}^3/\mathrm{mol} \cdot \mathrm{s}$ | Guibert et al. [128] |
| $E_p = 69172$ | $\mathrm{J}/\mathrm{mol}$ | Guibert et al. [128] |
| $A_t = 950410$ | $\mathrm{m}^3/\mathrm{mol} \cdot \mathrm{s}$ | $*$ |
| $E_t = 105018$ | $\mathrm{J}/\mathrm{mol}$ | Gee et al. [139] |
| $(-\Delta H_p) = 92048$ | $\mathrm{J}/\mathrm{mol}$ | Herrington and Hock [127] |

$*$ Calculated by using $k_p = 800 k_t$, when $T = 105\ ^\circ\mathrm{C}$

Table B.3: Kinetic parameters of KOH catalyzed propoxylation

# Appendix C

# Supplementary Information for Chapter 6

## C.1 Model Parameters

The order information used in the case study is given in Tab. C.1. The model parameters for the products and units of the mixed plant example is given in Tabs. D.1 and C.3. The task resource interaction parameters used in the RTN representations in Figs. 6.3- 6.6 are listed in Tab. D.2. The order value coefficients $c^{order}_{o,r,t}$ is calculated by

$$c^{order}_{o,r,t} = 1 + \frac{1}{10}\frac{t - E_o}{D_o - E_o}, \quad \forall (o,r) \in \Omega_{o,r}, E_o \leq t \leq D_o.$$

The orders are processed daily, where the earliest time $E_o$ and latest time $D_o$ are the first and last hour of each day, respectively. Finally, the penalty coefficients for violating the safety stock limits at the final time are $0.5$ for Prod. A to E.

| Day \ Prod. | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 70 | | | | |
| 2 | 70 | | 70 | | 70 |
| 3 | 70 | 70 | 70 | 70 | |
| 4 | 70 | | | 70 | |

Table C.1: Product order data for the case study examples

| Prod. | Product family | Int. F comp. | Batch size/time | $T_1$ rate max/min | $T_2$ rate max/min | $T_3/CU_{1/2}$ rate max/min | Inventory level initial/safety/max. |
|---|---|---|---|---|---|---|---|
| A | $F_1$ | 0.05 | 70/14 | 35/35 | 35/9.5 | 15/9.5 | 300/300/860 |
| B | $F_2$ | 0.01 | 70/14 | 35/35 | 35/9.5 | 15/9.5 | 108/108/310 |
| C | $F_1$ | 0.02 | 70/14 | 35/35 | 35/9.5 | 13/9.5 | 210/210/600 |
| D | $F_1$ | | 70/14 | 35/35 | 35/9.5 | 15/9.5 | 108/108/310 |
| E | $F_3$ | | 70/14 | 35/35 | 35/9.5 | 12/9.5 | 108/108/310 |
| F | | | 40/14 | | | | 10/ 0/100 |

* Time and processing rates in hours

Table C.2: Scheduling model parameters of the mixed plant: product information

| Unit. | Inventory lim.(/cost $c_r^{inv}$) | Changeover time/cost $c_i^{tr}$ | Processing capcity | Repln. time/thld | Discount factor $f_r$ |
|---|---|---|---|---|---|
| $BU_{1/2}$ | 70 | | | | |
| $T_1$ | 140/0.005 | | | | 0 |
| $T_2$ | 140/0.005 | | | | 0.3 |
| $CU_1$ | 0 | 6/5 | | | |
| $T_3$ | 80/0.005 | | | | 0.4 |
| $CU_2$ | 0 | 10/5 | 280 | 2/25% | |

* Time in hours
* The discount factors of different products are the same in the same buffer tank

Table C.3: Scheduling model parameters of the mixed plant: equipment information

| Batch processing unit (Fig. 6.3) | | | |
|---|---|---|---|
| $\mu_{1,BU,0} = -1$ | $\nu_{1,F,0} = -0.05$ | $\mu_{1,BU_s,14} = 1$ | $\nu_{1,A_{BU},14} = 1$ |
| $\nu_{2,A^o_{BU},0} = 1$ | $\mu_{2,BU_s,0} = -1$ | $\nu_{2,A_{BU},0} = -1$ | $\mu_{2,BU,0} = 1$ |
| **Buffer tank $T_1$ (Fig. 6.4)** | | | |
| $\mu_{1,T_1,0} = -1$ | $\nu_{1,1,A^o_{1,BU_1},0} = -1$ | $\nu_{1,2,A^o_{BU_2},0} = -1$ | $\nu_{1,3,A_{T_1},0} = -1$ |
| $\mu_{1,T_1,1} = 1$ | $\nu_{1,1,A_{T_1},0} = 1$ | $\nu_{1,2,A_{T_1},0} = 1$ | $\nu_{1,3,A^o_{T_1},0} = 1$ |
| $\alpha_{1,B_{T_1},0} = -140$ | $\alpha_{1,C_{T_1},0} = -140$ | $\alpha_{1,D_{T_1},0} = -140$ | $\alpha_{1,E_{T_1},0} = -140$ |
| $\alpha_{1,B_{T_1},1} = 140$ | $\alpha_{1,C_{T_1},1} = 140$ | $\alpha_{1,D_{T_1},1} = 140$ | $\alpha_{1,E_{T_1},1} = 140$ |
| **Continuous processing units (Fig. 6.5)** | | | |
| $\mu_{1,CU_1,0} = -1$ | $\nu_{1,A^o_{T_2},0} = -1$ | $\mu_{1,CU_1,1} = 1$ | $\nu_{1,A^o_{CU_1},0} = 1$ |
| $\mu_{2,T_3,0} = -1$ | $\nu_{2,1,A^o_{CU_1},0} = -1$ | $\nu_{2,2,A_{T_3},0} = -1$ | $\mu_{2,T_3,1} = 1$ |
| $\nu_{2,1,A_{T_3},0} = 1$ | $\nu_{2,2,A^o_{T_3},0} = 1$ | $\mu_{3,CU_2,0} = -1$ | $\nu_{3,1,A^o_{T_2},0} = -1$ |
| $\nu_{3,2,CAP,0} = -1$ | $\mu_{3,CU_2,1} = 1$ | $\nu_{3,1,A,0} = 1$ | $\mu_{4,CU_2,0} = -1$ |
| $\nu_{4,CAP,0} = -1$ | $\mu_{4,CU_2,2} = 1$ | $\mu_{4,CAP,2} = 280$ | $\alpha_{4,CAP,0} = -280$ |
| $\alpha_{4,CAP,2} = 280$ | | | |
| **Changeover tasks of $CU_1$ (Fig. 6.6)** | | | |
| $\mu_{1,CU_{1F_1},0} = -1$ | $\mu_{1,CU_{1F_2},6} = 1$ | $\mu_{2,CU_{1F_2},0} = -1$ | $\mu_{2,CU_{1F_1},6} = 1$ |
| $\mu_{3,CU_{1F_2},0} = -1$ | $\mu_{3,CU_{1F_3},6} = 1$ | $\mu_{4,CU_{1F_3},0} = -1$ | $\mu_{4,CU_{1F_2},6} = 1$ |
| $\mu_{5,CU_{1F_1},0} = -1$ | $\mu_{5,CU_{1F_3},6} = 1$ | $\mu_{6,CU_{1F_3},0} = -1$ | $\mu_{6,CU_{1F_1},6} = 1$ |

* Index of $\mu$ and $\alpha$: {task,resource,time}
* Index $\nu$: {task,extent,resource,time} (extent index omitted for tasks with single extent)

Table C.4: Task resource interaction parameters

# Appendix D

# Supplementary Information for Chapter 7

## D.1 Reactor Model Equations

The polymerization mechanism is the same for the three homopolymer products, which is similar to the polyol model shown in Tab. 4.1. Using the notation developed for moment models in Sec. 5.2.3, the moment balance equations are shown as follows:

$$\frac{\mathrm{d}X_0}{\mathrm{d}t} = -V^- k_i G_0 M, \tag{D.1.1a}$$

$$\frac{\mathrm{d}\lambda_0}{\mathrm{d}t} = V^- k_i G_0 M, \tag{D.1.1b}$$

$$\frac{\mathrm{d}\lambda_1}{\mathrm{d}t} = V^- (k_i G_0 + k_p \zeta_0) M \tag{D.1.1c}$$

$$\frac{\mathrm{d}\lambda_2}{\mathrm{d}t} = V^- (k_i G_0 + k_p (2\zeta_1 + \zeta_0)) M, \tag{D.1.1d}$$

$$\frac{\mathrm{d}Y_0}{\mathrm{d}t} = -V^- (k_i Q_0 + k_t n_c) M, \tag{D.1.1e}$$

$$\frac{\mathrm{d}\mu_0}{\mathrm{d}t} = V^- k_i Q_0 M, \tag{D.1.1f}$$

$$\frac{\mathrm{d}\mu_1}{\mathrm{d}t} = V^- (k_i Q_0 + k_p \nu_0) M, \tag{D.1.1g}$$

$$\frac{\mathrm{d}\mu_2}{\mathrm{d}t} = V^- (k_i Q_0 + k_p (2\nu_1 + \nu_0)) M. \tag{D.1.1h}$$

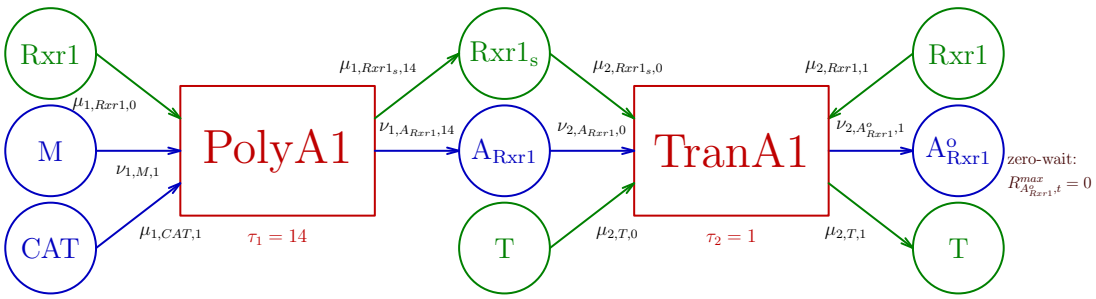



Figure D.1: RTN representation of Rxr1

For the algebraic equations, we obtain

$$\begin{aligned}
\mathrm{X}_0 n_c &= \mathrm{G}_0(\mathrm{X}_0 + \mathrm{Y}_0 + \lambda_0 + \mu_0), \\
\mathrm{Y}_0 n_c &= \mathrm{Q}_0(\mathrm{X}_0 + \mathrm{Y}_0 + \lambda_0 + \mu_0), \\
\lambda_k n_c &= \zeta_k(\mathrm{X}_0 + \mathrm{Y}_0 + \lambda_0 + \mu_0), \qquad k = \{0, 1\}, \\
\mu_k n_c &= \nu_k(\mathrm{X}_0 + \mathrm{Y}_0 + \lambda_0 + \mu_0), \qquad k = \{0, 1\}.
\end{aligned} \tag{D.1.2}$$

In this model, only the zeroth and first moments are computed. The rest of the reactor model equations and the optimization constraints are also similar to those developed in Sec. 5.2.3, except for only one monomer type is considered.

## D.2 RTN Representation of the Polymerization Process

The RTN representation of the process is similar to the RTN networks in Sec. 6.4. We show RTN network of Product A as an example.

### D.2.1 Polymerizaiton Reactors

The RTN representation of Rxr1 is shown in Fig. D.1. Two tasks are defined, corresponding to the polymerization operation (PolyA1) and the material transfer operation (TranA1). The values of the resource task interaction parameters are listed in Tab. D.2.

### D.2.2 Buffer Tank

the RTN representation for the intermediate storage tank is depicted in Fig. D.2. The buffering task spans a single time slot and seizes the equipment resource T during execution.

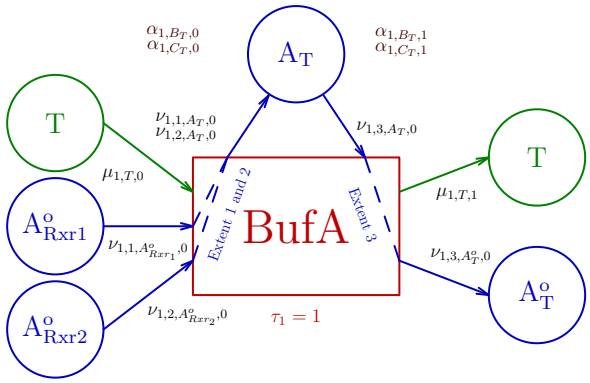
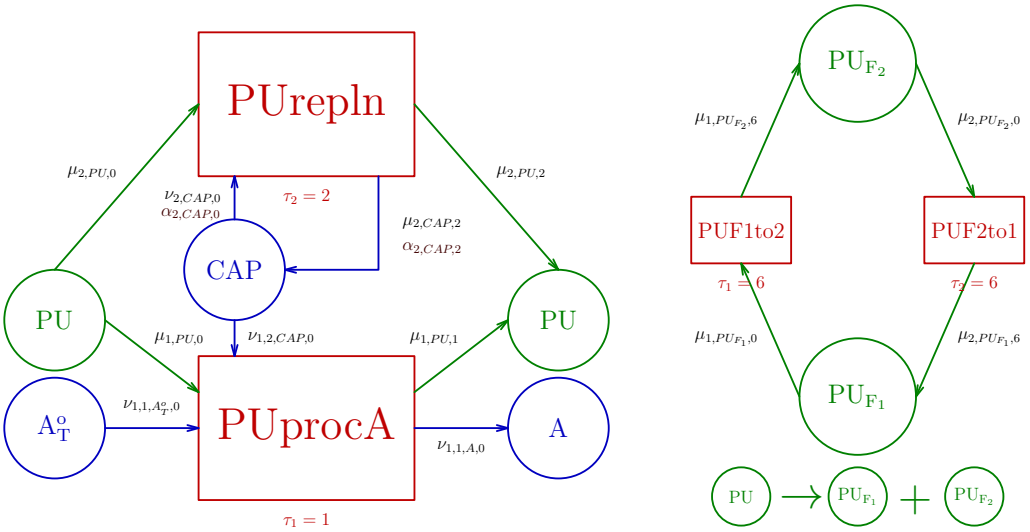



Figure D.2: RTN representation of the buffer tank




(a) Process tasks

(b) Changeover tasks

Figure D.3: RTN representation of the purification unit

### D.2.3 Purification Unit

We show the RTN representation of the continuous purification unit PU in Fig. D.3(a). Then two changeover tasks are defined to toggle the unit between the two states for producing products in different categories, shown in Fig. D.3(b).

| Prod. type | Batch size | Task length $m=1$ | Task length $m=2$ | Rxr setup time | Purification rate max/min | PU CAP cons. | Price (MU) | Order amount | Product family |
|---|---|---|---|---|---|---|---|---|---|
| A | 70 | 14 | | 1 | 17.5/5 | 1 | 0.3 | 140 | F1 |
| B | 70 | 10 | 11 | 1 | 17.5/5 | 1 | 0.3 | 280 | F2 |
| C | 70 | 8 | 9 | 1 | 23.3/5 | 1 | 0.3 | 140 | F1 |

* Time and processing rates in hours

Table D.1: Model parameters for the case study example: scheduling parameters

| | | | |
|---|---|---|---|
| Polymerization reactors (Fig. D.1) | | | |
| $\mu_{1,Rxr1,0} = -1$<br>$\nu_{2,A_{Rxr1},0} = -1$<br>$\mu_{2,Rxr1,1} = 1$ | $\nu_{1,M,1} = -1$<br>$\mu_{2,Rxr_s,0} = -1$<br>$\mu_{2,T,1} = 1$ | $\mu_{1,Rxr1_s,14} = 1$<br>$\mu_{2,T,0} = -1$ | $\nu_{1,A_{Rxr1},14} = 1$<br>$\nu_{2,A^o_{Rxr1},1} = 1$ |
| Buffer tank (Fig. D.2) | | | |
| $\mu_{1,T,0} = -1$<br>$\mu_{1,T,1} = 1$<br>$\alpha_{1,B_T,0} = -140$ | $\nu_{1,1,A^o_{1,Rxr1},0} = -1$<br>$\nu_{1,1,A_T,0} = 1$<br>$\alpha_{1,C_T,0} = -140$ | $\nu_{1,2,A^o_{Rxr2},0} = -1$<br>$\nu_{1,2,A_T,0} = 1$<br>$\alpha_{1,B_T,1} = 140$ | $\nu_{1,3,A_T,0} = -1$<br>$\nu_{1,3,A^o_T,0} = 1$<br>$\alpha_{1,C_T,1} = 140$ |
| Purification unit (Fig. D.3(a)) | | | |
| $\mu_{1,PU,0} = -1$<br>$\nu_{1,1,A,0} = 1$<br>$\mu_{2,CAP,2} = 280$ | $\nu_{1,1,A^o_T,0} = -1$<br>$\mu_{2,PU,0} = -1$<br>$\alpha_{2,CAP,0} = -280$ | $\nu_{1,2,CAP,0} = -1$<br>$\nu_{2,CAP,0} = -1$<br>$\alpha_{2,CAP,2} = 280$ | $\mu_{1,PU,1} = 1$<br>$\mu_{2,PU,2} = 1$ |
| Changeover tasks of PU (Fig. D.3(b)) | | | |
| $\mu_{1,PU_{F_1},0} = -1$ | $\mu_{1,PU_{F_2},6} = 1$ | $\mu_{2,PU_{F_2},0} = -1$ | $\mu_{2,PU_{F_1},6} = 1$ |

* Index of $\mu$ and $\alpha$: {task,resource,time} (task mode index not included for Product A)
* Index $\nu$: {task,extent,resource,time} (extent index omitted for tasks with single extent)

Table D.2: Task resource interaction parameters