On Synthesis, Design and Resource Optimization in Multipurpose Batch Plants

Esmael Seid

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

On Synthesis, Design and Resource Optimization in Multipurpose Batch Plants

by

Esmael Seid

A thesis submitted in partial fulfillment of the requirements for the degree

PhD: Chemical Engineering

In the

Faculty of Engineering, the Built Environment and Information Technology

University of Pretoria

Pretoria

Supervisor: Prof. Thokozani Majozi

2013

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Synopsis

In recent years, batch processes have been getting more attention due to their suitability for the production of small volume, high value added products. The flexibility of batch plants allows the production of different products within the same facility which mandates equipment sharing. Batch manufacturing is typically used in the pharmaceutical, polymer, food and specialty chemical industries as demands for such products are highly seasonal and are influenced by changing markets. Despite the advantage of batch plants being flexible, they also pose a challenging task to design, synthesize and operate, compared to their continuous counterparts. The profitability of these batch plants is highly dependent on the way the synthesis, design and operation is optimized. Since different types of resources (raw materials, equipment, utilities and manpower) need to be shared by a number of process operations to produce a variety of products, modeling and optimizing the design and operation of batch plants are important for economic benefits.

The growing awareness of civil society for the environment and the resulting regulations introduced by national states have resulted in chemical industries considering process integration to reduce their energy and process water requirements. Energy optimization and the optimization of water use have mainly been treated as separate problems in literature. The batch production schedules resulting from each of these formulations do not guarantee that the plant is operated optimally. Consequently, it is required to develop a formulation that caters for opportunities that exist for both wastewater minimization and energy integration. This may result in production schedules that improve the operation of the batch plant when compared to optimizing water and energy separately.

Presented in this thesis is a mathematical technique that addresses optimization of both water and energy, while simultaneously optimizing the batch process schedule. The scheduling framework used in this study is based on the formulation by Seid and Majozi (2012). This formulation has been shown to result in a significant reduction of computational time, an improvement of the objective function and leads to fewer time points required to solve the scheduling problem. The objective is to improve the profitability of the plant by minimizing wastewater generation and utility usage. From a case study it was found that through only applying water integration the total cost is reduced by 11.6%, by applying only energy integration the total cost is reduced by 29.1% and by applying both energy and water integration the total cost is reduced by 34.6%. This indicates that optimizing water and energy integration in the same scheduling framework will reduce the operating cost and environmental impact significantly.

This thesis also presents a mathematical model for design and synthesis of batch plants. The conceptual design problem must determine the number and capacity of the major processing equipment items, pipe connections and storage tanks so as to meet production objectives at the lowest possible capital and operating cost. A recent robust scheduling model based on continuous-time representation is used as a platform for the synthesis and design problem. An improved objective value (revenue) of 228.6% is obtained by this work compared to the recent published models for the design and synthesis problem. Compared with other formulations, the formulation presented in this thesis gives a smaller size mathematical model that required less binary variables, continuous variables and constraints. The presented model also considers costs that arise from the pipe network and consequently, determines the optimal pipe network which should exist between different pieces of equipment.

Finally, the medium-term scheduling problem for a multiproduct batch plant is addressed. The intractability of the short-term scheduling models when directly applied to the medium-term scheduling problems is solved by applying a decomposition method. The decomposition method has two level mathematical models. The first level determines the type of products and their amount to be produced in each scheduling subproblem to satisfy the market requirement. The second level determines the detailed sequencing of tasks for the tractable size of the subproblems. The recently published robust short-term scheduling model based on continuous time is extended for solving the scheduling supbroblems of the second level decomposition model. The model is applied in solving the medium-term scheduling problem of a pharmaceutical facility specializing in animal vaccines using the actual plant data. The model effectively solved a makespan minimization problem for the medium-term scheduling horizon of almost 13 weeks.

I, Esmael Seid

Student number: u29439592

Topic of work: On Synthesis, Design and Resource Optimization in Multipurpose Batch Plants

Declaration

I understand what plagiarism is and am aware of the university policy in this regard.

1. I declare that this dissertation is my own, original work. Where someone else's work was used (either from a printed source, the internet or any other source) due acknowledgement was given and a reference was made according to the departmental requirements.
2. I have not used another student's past written work to hand in as my own.
3. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

Signature ---------------------------

Acknowledgments

First and foremost I would like to thank my supervisor Prof Thokozani Majozi for his willingness to accept and supervise my PhD study at the University of Pretoria. He is my promoter, reliable mentor, guide, and instructor. His insightful critical comments, never ending patience and encouragements have been invaluable to make my work up to the standard.

I am also grateful to SUSPSE research group at the University of Pretoria for their helpful contribution to my work and for our best friendship.

I would like to thank Dr Abubeker and his wife Amina for their guardian, mentor, support and useful discussion that I had on social life. My gratitude also goes to Mesay, Mehratu, Dr Adem and Melhik for having constructive support in our daily life in general.

Finally but most importantly I would like to salute my dear and beloved mother Sada Hassen, my father, sisters and brothers for their support in all direction to upgrade myself to a higher level. I truly could not accomplish this without them.

TABLE OF CONTENTS

List of abbreviations

<u>Abbreviations Full name</u>

State sequence network (SSN)

Resource task network (RTN)

State task network (STN)

Maximal state task network (m-STM)

Mixed integer linear programming (MILP)

Mixed integer nonlinear programming (MINLP)

Zero waits (ZW)

Fixed intermediate storage (FIS)

No intermediate storage (NIS)

List of figures

List of tables

1

CHAPTER 1

Introduction

1.1 Background

Any process which is a consequence of discrete tasks that have to follow a predefined sequence from raw materials to final products is a batch process. This predefined sequence is commonly known as a recipe. Batch processing plants are attractive due to their suitability for the production of small volume, high value added products, which are becoming increasingly important due to fast market changes. Batch manufacturing is typically used in the pharmaceutical, polymer, food and specialty chemical industries, because it provides the necessary flexibility to accommodate various production requirements using the same processing facility. The batch process industries can be categorized into multipurpose and multiproduct batch plants. In general, batch processes are characterized as follows (Liu, 1996):

- Manufacturing processes involving a set of operations that are executed independently and in batches
- Sharing of resources (such as operators, steam, electricity or auxiliary equipment)
- Presence of intermediate storage to separate operations and mitigate the effects of process variations or upset
- Multipurpose equipment (e.g. a piece of equipment may be used for either. processing different tasks or as a storage unit)

- Flexibility in configuration (since the equipment can often be connected in different ways)
- Decision-dependent set-up costs are involved
- High quality specifications

Generally batch operations present more challenging environmental problems. While the size of the processing plants is generally smaller, batch operations typically exhibit a higher ratio of waste by-products produced per unit of product. Coupled with the wastewater problem is the issue of greenhouse gas emissions due to the energy usage. Greater public awareness of the impact of industrial pollution and more stringent environmental regulations have now encouraged research work in energy and water saving measures for more sustainable operations (Halim and Srinivasan, 2011). Consequently, chemical industries have been changing significantly in design, of operating conditions and optimal usage of energy and water.

Most engineers believe that improving process efficiencies, optimal design and synthesis and minimization of energy and wastewater will be the main focus in the near future to solve the environmental problems and increase the profitability of batch plants. The substantial advancement in modern computers gives hope to deal with large size and more complex problems in the chemical industry by using mathematical optimization techniques. The subsections below briefly describe the challenges and published literature available in solving the aforementioned problems.

1.1.1 Energy integration

Energy saving in continuous processes so far is effectively applied by using the well-known technique of pinch analysis. However, energy saving opportunities for batch processes have been largely neglected, because batch processes are in general less energy intensive than continuous bulk systems. Nevertheless, the utility requirements for the food industry, breweries, dairies, meat processing facilities, biochemical plants and agrochemical facilities contribute largely to their overall cost (Majozi, 2009; Halim and Srinivasan, 2009). Consequently, it is worth studying since existing batch processes often have little or no heat recovery and thus there are more energy saving opportunities. Both pinch analysis and practical heat recovery projects are much more difficult than for continuous processes for a number of reasons: Many streams are present for only certain time periods, which make time an additional variable in the heat integration model. Also, they may not run between constant temperatures. Heating or cooling is supplied by an external jacket or internal coils, and the vessel contents gradually change in temperature. Nevertheless, despite these difficulties, considerable research has been done to address the challenges by continually improving the mathematical models to achieve better energy saving and reduce the computational effort required.

The methodologies for incorporating heat integration in batch scheduling can be divided into sequential and simultaneous approaches. The former involves decomposing the problem into two: the scheduling part has to be specified a priori or solved first followed by heat integration synthesis to simplify the model (Vaselenak et al., 1986; Vaklieva-Bancheva et al. 1996; Uhlenbruck et al., 2000; Halim and Srinivasan, 2009). For a more optimal solution, scheduling and heat integration may be combined into an overall problem and solved simultaneously (Papageorgiou et al., 1994; Georgiadis and Papageorgiou, 2001; Stamp and Majozi, 2011).

1.1.2 Water reuse network synthesis

Following the concept of heat integration, water reuse network synthesis has been developed to address the increasing concern on wastewater generation. The first step in accomplishing wastewater minimization is through water targeting. Targeting determines the minimum freshwater and wastewater flowrates that can be achieved after water reuse/recycle opportunities within a processing plant have been employed. This is followed by the synthesis of the water network subject to the contaminant mass load, maximum allowable contaminant concentration of process units, and timing of the operations. The methodologies developed for wastewater minimization in batch plants can be broadly categorized into graphical and mathematical based techniques.

In the graphical techniques (Wang and Smith, 1995) the problem is divided into concentration intervals and time subintervals with the boundaries of the intervals determined by the end points of the individual processes. Targeting is performed in each concentration interval by reusing water which is available in its own time intervals, where possible. Any surplus water could be reused in subsequent time intervals in the same concentration interval or could be stored for reuse in later concentration intervals.

When the opportunities for recycle/reuse are depleted, freshwater is used. The advantage of this methodology is that it can easily determine the wastewater reuse bottleneck. However, this methodology is constrained to two dimensions and thus cannot be used to solve process streams with multiple contaminants. Consequently, much of the research has been directed towards developing mathematical optimization techniques (Kim and Smith, 2004; Majozi, 2005; Shoaib et al., 2008; Majozi and Gouws, 2009; Li et al., 2010; Adekola and Majozi, 2011). The mathematical programming techniques have the following advantages:

- Allow the opportunity to optimize different functions
- Can be used to solve multicontaminant systems
- Can handle restrictions from forbidden matches and flowrates
- Production can also be optimized simultaneously

1.1.3 Design and synthesis of batch plants

Continual developments in the design and synthesis of batch plants have been carried out in the last two decades. These developments were focused essentially on model enhancements that originate higher model accuracy allowing better description of real problems. The analysis of literature on the grassroots design of multipurpose batch plants is structured in two main subdivisions. First, where the simple choice of equipment and associated scheduling are considered—"basic design", (Vaselenak et al., 1987; Faqir and Karimi, 1989; Xia and Macchietto, 1997; Lin and Floudas 2001; Castro et al., 2005); second, where further plant aspects are incorporated into the design model such as uncertainty, plant topology, heat integration and environmental aspects—"extended design" (Barbosa-Povoa and Macchietto, 1994; Pinto et al., 2003; Patziatsis, et al., 2005; Pinto et al., 2008).

1.2 Motivation

Despite the benefits, the inherent nature of batch processes complicates the simultaneous optimization of the schedule, along with energy and water consumption. Due to complexities, problems entailing wastewater minimization with multiple contaminants and energy integration for multi-purpose batch plants in literature have been inadequately addressed with rigorous and efficient methodologies. Further, there is lack of a unified methodology that solves the problem of scheduling, heat integration and water minimization within a single framework (Halim and Srinivasan, 2011). A critical review on the design and retrofit of batch plants by Barbosa-Póvoa (2007) showed that good solution algorithms and a comparative computational investigation on the different approaches proposed to solve the design problem would be of immense value.

1.3 Aim

The goal of the investigation in this thesis is to develop a formulation that addresses the optimization of both water and energy, while simultaneously optimizing the batch process schedule. Moreover, the investigation also addresses the gap in relation to good solution algorithms for design and synthesis problems for multipurpose batch plants by using the robust scheduling framework recently developed by Seid and Majozi (2012) and present the comparative computational results. This work also explores the development of a medium-term scheduling technique to a pharmaceutical industry by using real industrial data.

1.4 Thesis structure

The subsequent part of the thesis is organized as follows: Chapter 2 presents a new formulation for energy integration for multipurpose batch plants that considers both direct and indirect energy integration accompanied by the comparative computational results. Chapter 3 discusses the novel simultaneous optimization of energy, water and production schedule within the same framework. The development of an efficient model for design and synthesis problems for multipurpose batch plants that considers plant topology is presented in Chapter 4. Chapter 5 discusses the medium-range production schedule for multiproduct batch plants. Finally, Chapter 6 concludes the findings of this investigation with recommendations and further work that needs to be addressed.

References

Adekola, O., Majozi, T., 2011. Wastewater minimization in multipurpose batch plants with a regeneration unit: Multiple contaminants. *Comput. Chem. Eng*. 35, 2824-2836.

Barbosa-Póvoa, A.P., Macchietto, S., 1994. Detailed design of multipurpose batch plants. *Comput. Chem. Eng*. 18(11/12), 1013–1042.

Barbosa-Póvoa, A.P.F.D., 2007. A critical review on the design and retrofit of batch plants. *Computers & Chemical Engineering*. 31 (7), 833–855.

Castro, P.M., Barbosa-Povoa, A.P., Novais, A.Q., 2005. Simultaneous design and scheduling of multipurpose plants using resource task network based continuous-time formulations. *Ind. Eng. Chem. Res*. 44(2), 343–357.

Faqir, N.M., Karimi, I.A., 1989. Optimal-design of batch plants with single production routes. *Ind Eng Chem Res*. 28(8), 1191–1202.

Georgiadis, M.C., Papageorgiou, L.G., 2001. Optimal scheduling of heat-integrated multipurpose plants under fouling conditions. *Appl. Therm. Eng*. 21, 1675–97.

Halim, I., Srinivasan, R., 2009. Sequential methodology for scheduling of heat-integrated batch plants. *Ind. Eng. Chem*. Res. 48, 8551–65.

Halim, I., Srinivasan, R., 2011. Sequential methodology for integrated optimization of energy and water use during batch process scheduling, *Comput. Chem. Eng*. 35, 1575-1597.

Kim, J.K., Smith, R., (2004). Automated design of discontinous water system. *Process Saf. environ.* 82(3), 238-248.

Li, L.J., Zhou, R.J., Dong, H.G., 2010. State-time-space superstructure based MINLP formulation for batch water allocation network design. *Ind. Eng. Chem. Res*. 49, 236-251.

Lin, X., Floudas, C.A., 2001. Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. *Comput. Chem. Eng*. 25665–674.

Liu, R., 1996. A framework for operational strategies for pipe less plants. Ph.D. Thesis, Department of Chemical Engineering, The University of Leeds, pp. 26-28.

Majozi, T., 2005. Wastewater minimization using central reusable water storage in batch processes. *J. Clean.Prod*. 13(15), 1374-1380.

Majozi, T., 2009. Minimization of energy use in multipurpose batch plants using heat storage: an aspect of cleaner production. J. Clean. Prod. 17, 945–50.

Majozi, T., Gouws, J.F., 2009. A mathematical optimisation approach for wastewater minimization in multipurpose batch plants: Multiple contaminants. *Comp. Chem. Eng*. 33, 1826-1840.

Papageorgiou, L.G., Shah, N., Pantelides, C.C., 1994. Optimal scheduling of heat-integrated multipurpose plants. *Ind. Eng. Chem. Res*. 33, 3168–186.

Patziatsis, D.I., Xu, G., Papageorgiou, L.G., 2005. Layout Aspects of pipeless batch plants. *Ind. Eng. Chem. Res*. 44(15), 5672–5679.

Pinto, T., Barbosa-Póvoa, A.P., Novais, A.Q., 2003. Optimal design of heat integrated scheduling and design of multipurpose batch plants. *Ind. Eng. Chem. Res*. 42(4), 836–846.

Pinto, T., Barbosa-Póvoa, A.P., Novais, A.Q., 2008. Design of multipurpose batch plants: A comparative analysis between the STN, m-STN, and RTN representations and formulations. *Ind. Eng. Chem. Res*. 47, 6025-6044.

Seid, R., Majozi, T., 2012. A robust mathematical formulation for multipurpose batch plants**.** *Chem. Eng. Sci*. 68, 36–53.

Shoaib, A.M., Aly, S.M., Awad, M.E., Foo, D.C.Y., El Halwagi, M.M., 2008. A hierarchical approach for the synthesis of batch water network. *Comput. Chemical. Eng*. 32, 530–539.

Stamp, J., Majozi, T., 2011. Optimal heat storage design for heat integrated multipurpose batch plants. *Energy*. 36(8), 1-13.

Uhlenbruck, S., Vogel, R., Lucas, K., 2000. Heat integration of batch processes. *Chem. Eng. Technol*. 23, 226–9.

Vaklieva-Bancheva, N., Ivanov, B.B., Shah, N., Pantelides, C.C., 1996. Heat exchanger network design for multipurpose batch plants. *Comput. Chem. Eng*. 20, 989–1001.

Vasenlenak, J.A, Grossmann, I.E, Westerberg, A.W., 1986. Heat integration in batch processing. *Ind. Eng. Chem. Process Des. Dev*. 25, 357–66.

Vasenlenak, J.A., Grossmann, I.E., Westerberg, A.W., 1987. An embedding formulation for the optimal scheduling and design of multipurpose batch plants. *Ind. Eng. Chem. Res*. 26(1), 139–148.

Wang, Y. P., Smith, R., 1994. Wastewater minimization. *Chem. Eng. Sci.* 49(7), 981–1006.

Xia, Q.S., Macchietto, S., 1990. Design and synthesis of batch plants MINLP solution based on a stochastic method. *Comput.Chem Eng*. 21, S697–S702.

2

CHAPTER 2

Heat Integration in Multipurpose Batch Plants Using a Robust Scheduling Framework

Abstract

Energy saving is becoming increasingly important in processing facilities. Although most common in continuous processes, energy integration has also become essential for batch processes. Batch plants have become more popular than ever in the processing environment. This is due to their inherent flexibility and adaptability to market conditions. This flexibility although may lead to complexities such as the need to schedule process tasks. Many current methods for energy saving in batch plants use a sequential methodology where the schedule is solved first and heat integration is then performed. This may lead to suboptimal results. In this chapter, the heat integration model is built upon the robust scheduling framework of Seid and Majozi (2012). This scheduling formulation has proven to lead to better results in the form of a better objective value, fewer required time points and reduced computational time. This is important as inclusion of heat integration into a scheduling model invariably complicates the solution process. The improved scheduling model allows the consideration of industrial size problems and allows the simultaneous optimization of both the process schedule and energy usage.

Both direct and indirect heat integration are considered as well as fixed and variable batch sizes.

Keywords: Heat integration; Scheduling; Multipurpose batch plant; Energy optimization.

2.1 Introduction

Batch operations, even though they are becoming increasingly popular, are generally run on a smaller scale compared to continuous operations and utility requirements are therefore considered less significant. However, the utility requirements for the food industry, breweries, dairies, meat processing facilities, biochemical plants and agrochemical facilities contribute largely to their overall cost (Knopf et al., 1982; Mignon and Hermia, 1993; Tokos et al., 2010; Atkins et al., 2010; Fritzson and Berntsson, 2006; Boyadjiev et al., 1996; Rašković et al., 2010; Majozi, 2009). Energy savings have often been neglected in batch processes in the past and so large percentage savings are possible.

Many heat integration techniques are applied to predefined schedules and suboptimal results are obtained. Early work on heat integration in batch processes (Vaselenak et al., 1986) explored heat exchange between hot and cold vessels requiring cooling and heating, respectively, in order to reduce utility consumption. Vaklieva-Bancheva et al. (1996) considered direct heat integration with the objective of minimizing total costs. The resulting overall formulation was a mixed integer linear programming (MILP) problem, solved to global optimality, although only specific pairs of units were allowed to undergo heat integration. Instead of analyzing batch streams from a thermodynamic perspective, Uhlenbruck et al. (2000) proposed first synthesizing all possible heat exchanger networks using direct heat integration. The given schedule was divided into time and temperature intervals. One hot stream was allowed to exchange with one cold stream via a countercurrent heat exchanger. The heat recovery was improved further by including matches of residual and previously unmatched streams. The method could not achieve the thermodynamic optimum. Halim and Srinivasan (2009) discussed a sequential method using direct heat integration. A number of optimal schedules with minimum makespan were found and heat integration analysis was performed on each. The schedule with minimum utility requirement was chosen as the best. It was argued that sequential procedures could lead to a higher number of practically implementable networks with an optimal schedule and are also more suitable for complex problems.

For a more optimal solution, scheduling and heat integration may be combined into an overall problem. Papageorgiou et al. (1994) embedded a heat integration model within the scheduling formulation of Kondili et al. (1993). Opportunities for both direct and indirect heat integration were considered as well as possible heat losses from the heat storage tank. The operating policy, in terms of heat integrated or standalone, was predefined for tasks. This work was extended by Georgiadis and Papageorgiou (2001) to consider fouling of heat exchange units and the associated cleaning schedules and costs. Adonyi et al. (2003) used the “S-Graph” scheduling approach (Sanmartí et al., 1998) and incorporated one to one direct heat integration. Heat integration was greatly improved with a small compromise in minimal makespan. Majozi (2006) presented a direct heat integration formulation based on the state sequence network (SSN) (Majozi and Zhu, 2001) and an unevenly discretized time horizon. The heat integration formulation as given was, however, more suited to multiproduct applications rather than multipurpose facilities. This work was later extended (Majozi, 2009) to include heat storage for indirect heat integration. The heat storage capacity and initial storage temperature were, however, predefined parameters. Although this led to a mixed integer linear programming (MILP) formulation, suboptimal results were obtained. Chen and Chang (2009) extended the work of Majozi (2006) to periodic scheduling, based on the resource task network (RTN). The resultant direct heat integration formulation was a MILP problem. The SSN formulation of Majozi (2006) used fewer binary variables than the RTN approach for the heat integrated short term scheduling case, while achieving the same objective value. However, for the periodic case, all heat sources and sinks operated in integrated mode making the process more economical. Stamp and Majozi (2011) presented a simultaneous scheduling and heat integration formulation suitable for multipurpose batch plants based on the work of Majozi (2009). A more detailed review on heat integration in batch plants can be obtained in the review by Fernández et al. (2012).

The proposed model presented in this chapter aims to improve the efficiency in simultaneous scheduling and heat integration in multipurpose batch plants. This is done by using the robust scheduling formulation by Seid and Majozi (2012) which has proven to require fewer time points and reduced computational time compared to other models and may also lead to an improved objective value. This formulation can be found in the Appendix. The rest of this chapter is organized as follows: the problem statement and objectives are given in the next section. The mathematical model is then discussed in Section 2.3. The model is then applied to two literature examples and an industrial case study in Section 2.4. Conclusions are then drawn highlighting the value of the contribution in Section 2.5.

2.2 Problem statement and objectives

The problem addressed in this work can be stated as follows:

Given:

(i) Production scheduling data, including equipment capacities, durations of tasks, time horizon of interest, product recipes, cost of starting materials and selling price of final products

(ii) Hot duties for tasks requiring heating and cold duties for tasks that require cooling

(iii) Costs of hot and cold utilities

(iv) Operating temperatures of heat sources and heat sinks

(v) Minimum allowable temperature differences, and

(vi) Design limits on heat storage

Determine:

(a) An optimal production schedule where the objective is to maximize profit, defined as the difference between revenue and the cost of hot and cold utilities.

(b) The size of heat storage available as well as the initial temperature of heat storage.

2.3 Mathematical model

The SSN recipe representation and an uneven discretization of the time horizon were used. The scheduling model by Seid and Majozi (2012) was used since it has proven to result in fewer binary variables, CPU time and a better optimal objective value compared to other scheduling models.

The mathematical model is based on the superstructure in Figure 2.1. Each task may operate using either direct or indirect heat integration. Tasks may also operate in standalone mode, using only external utilities. This may be required for control reasons or when thermal driving forces or time do not allow for heat integration. If either direct or indirect heat integration is not sufficient to satisfy the required duty, external utilities may make up for any deficit.

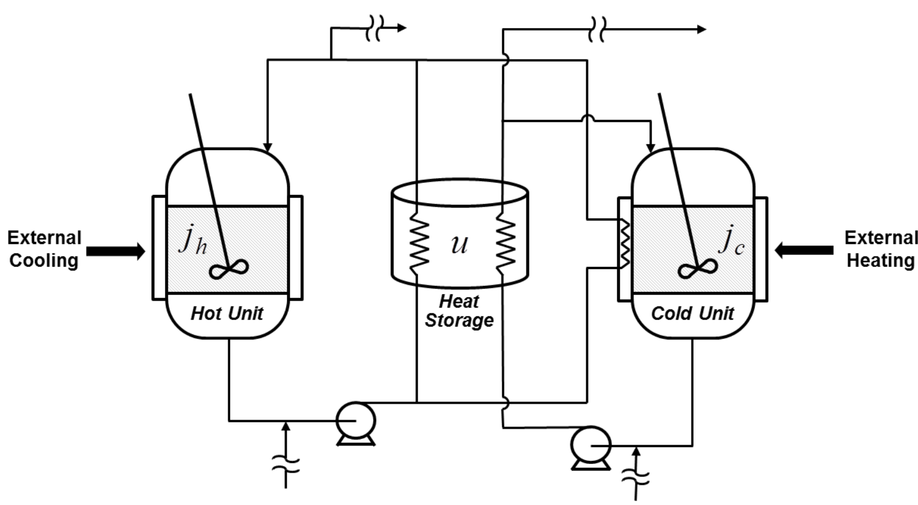


Figure 2.1. Superstructure for mathematical model.

The mathematical model comprises the following constraints: In addition to the necessary short term scheduling constraints by Seid and Majozi (2012), Constraints (2-1) to (2-45) constitute the heat integration model, useful for multipurpose batch processes with fixed and variable batch sizes. Both direct and indirect heat integration are considered.

Constraints (2-1) and (2-2) are active simultaneously and ensure that one hot unit will be integrated with one cold unit when direct heat integration takes place, in order to simplify operation of the process. Also, if two units are to be heat integrated at a given time point, they must both be active at that time point. However, if a unit is active, it may operate in either integrated or standalone mode.

$$\sum_{s_{injc}} x\left(s_{injc}, s_{injh}, p\right) \le y\left(s_{injh}, p\right), \quad \forall\ p \in P, \quad s_{injh} \in S_{inJ_h} \tag{2-1}$$

$$\sum_{s_{injh}} x\left(s_{injc}, s_{injh}, p\right) \le y\left(s_{injc}, p\right), \quad \forall\ p \in P, \quad s_{injc} \in S_{inJ_c} \tag{2-2}$$

Constraint (2-3) ensures that only one hot or cold unit is heat integrated with one heat storage unit at any point in time. This is to simplify and improve operational efficiency in the plant.

$$\sum_{s_{injc}} z\left(s_{injc}, u, p\right) + \sum_{s_{injh}} z\left(s_{injh}, u, p\right) \le 1, \ \forall\ p \in P, \quad u \in U \tag{2-3}$$

Constraints (2-4) and (2-5) ensure that a unit cannot simultaneously undergo direct and indirect heat integration. This condition simplifies the operation of the process.

$$\sum_{s_{injh}} x\left(s_{injc}, s_{injh}, p\right) + z\left(s_{injc}, u, p\right) \le 1, \ \forall\ p \in P, \quad s_{injc} \in S_{inJ_c}, \quad u \in U \tag{2-4}$$

$$\sum_{s_{injc}} x\left(s_{injc}, s_{injh}, p\right) + z\left(s_{injh}, u, p\right) \le 1, \ \forall\ p \in P, \quad s_{injh} \in S_{inJ_h}, \quad u \in U \tag{2-5}$$

Constraints (2-6) and (2-7) quantify the amount of heat received from or transferred to the heat storage unit, respectively. There will be no heat received or transferred if the binary variable signifying use of the heat storage vessel, $z\left(s_{inj}, u, p\right)$, is zero.

$$Q(s_{inj_c}, u, p) = W(u) c_p (T_0(u, p) - T_f(u, p)) z(s_{inj_c}, u, p), \ \forall \ p \in P, \quad s_{inj_c} \in S_{inJ_c}, \ u \in U \tag{2-6}$$

$$Q(s_{inj_h}, u, p-1) = W(u) c_p (T_f(u, p) - T_0(u, p)) z(s_{inj_h}, u, p), \ \forall \ p \in P, \quad s_{inj_h} \in S_{inJ_h}, \ u \in U \tag{2-7}$$

Constraint (2-8) ensures that the final temperature of the heat storage fluid at any time point becomes the initial temperature of the heat storage fluid at the next time point. This condition will hold regardless of whether or not there was heat integration at the previous time point.

$$T_0(u, p) = T_f(u, p-1), \ \forall \ p \in P, \quad u \in U \tag{2-8}$$

Constraints (2-9) and (2-10) ensure that the heat storage temperature does not change if there is no heat integration with the heat storage unit. M is any large number equivalent to the maximum temperature from all hot tasks, thereby resulting in an overall "Big M" formulation. If either $z(s_{inj_c}, u, p)$ or $z(s_{inj_h}, u, p)$ is equal to one, Constraint (2-9) and Constraint (2-10) will be redundant. However, if these two binary variables are both zero, the initial temperature at the previous time point will be equal to the final temperature at the current time point if heat losses are ignored.

$$T_0(u, p) \le T_f(u, p) + M\left(\sum_{s_{inj_c}} z(s_{inj_c}, u, p-1) + \sum_{s_{inj_h}} z(s_{inj_h}, u, p-1)\right), \ \forall \ p \in P, \quad u \in U \tag{2-9}$$

$$T_0(u, p) \ge T_f(u, p) - M\left(\sum_{s_{inj_c}} z(s_{inj_c}, u, p-1) + \sum_{s_{inj_h}} z(s_{inj_h}, u, p-1)\right), \ \forall \ p \in P, \quad u \in U \tag{2-10}$$

Constraint (2-11) ensures that minimum thermal driving forces are obeyed when there is direct heat integration between a hot and a cold unit.

$$T(s_{inj_h}) - T(s_{inj_c}) \ge \Delta T^{\min} - M(1 - x(s_{inj_c}, s_{inj_h}, p)), \qquad \forall \ p \in P, \ s_{inj_c}, s_{inj_h} \in S_{inJ} \tag{2-11}$$

Constraints (2-12) and (2-13) ensure that minimum thermal driving forces are obeyed when there is heat integration with the heat storage unit. Constraint (2-12) applies for heat integration between heat storage and a heat sink, while constraint (2-13) applies for heat integration between heat storage and a heat source.

$$T_f(u,p)-T(s_{inj_c})\geq \Delta T^{\min}-M(1-z(s_{inj_c},u,p-1)),\quad \forall\ p\in P,\ s_{inj_c}\in S_{inj_c},\quad u\in U \qquad (2\text{-}12)$$

$$T(s_{inj_h})-T_f(u,p)\geq \Delta T^{\min}-M(1-z(s_{inj_h},u,p-1)),\quad \forall\ p\in P,\ s_{inj_h}\in S_{inj_h},\quad u\in U \qquad (2\text{-}13)$$

Constraint (2-14) states that the cooling of a heat source will be satisfied by either direct or indirect heat integration as well as external utility if required.

$$(E(s_{inj_h})y(s_{inj_h},p)+\xi(s_{inj_h},p)mu(s_{inj_h},p))(tp(s_{inj_h},p)-tu(s_{inj_h},p))=Q(s_{inj_h},u,p)+cw(s_{inj_h},p)$$
$$+\sum_{s_{inj_c}} xx(s_{inj_h},s_{inj_c},p),\ \forall\ p\in P,\quad s_{inj_h}\in S_{inJ_h},\quad u\in U \qquad (2\text{-}14)$$

Constraint (2-15) ensures that the heating of a heat sink will be satisfied by either direct or indirect heat integration as well as external utility if required.

$$(E(s_{inj_c})y(s_{inj_c},p)+\xi(s_{inj_c},p))*tp(s_{inj_c},p)-tu(s_{inj_c},p))=Q(s_{inj_c},u,p)+st(s_{inj_c},p)$$

$$+\sum_{s_{inj_h}} xx(s_{inj_h},s_{inj_c},p)\qquad \forall\ p\in P,\quad s_{inj_h}\in S_{inJ_h},\quad u\in U \qquad (2\text{-}15)$$

Constraint (2-16) states that the amount of heat exchanged between the hot and cold unit is limited by the total duration of the cold unit.

$$xx(s_{inj_h},s_{inj_c},p)\leq (E(s_{inj_h})y(s_{inj_h},p)+\xi(s_{inj_h},p)mu(s_{inj_h},p))(tp(s_{inj_c},p)-tu(s_{inj_c},p)),$$

$$\forall\quad p\in P,\ s_{inj_c}\in S_{inJ_c},\ s_{inj_h}\in S_{inJ_h} \qquad (2\text{-}16)$$

Constraint (2-17) ensures that the amount of heat transferred from the hot unit to the cold unit is limited by the duration of the hot unit.

$$xx(s_{inj_h},s_{inj_c},p)\leq (E(s_{inj_c})y(s_{inj_c},p)+\xi(s_{inj_c},p)mu(s_{inj_c},p))(tp(s_{inj_h},p)-tu(s_{inj_h},p)),$$
$$\forall\ p\in P,\quad s_{inj_c}\in S_{inJ_c},\ s_{inj_h}\in S_{inJ_h} \qquad (2\text{-}17)$$

Constraint (2-18) states that the amount of heat exchanged between the hot and cold unit takes a nonzero value if the binary variable $x(s_{inj_c},s_{inj_h},p)$ for direct heat integration takes a value of one, otherwise it becomes zero.

$$xx(s_{inj_h},s_{inj_c},p)\leq Mx(s_{inj_c},s_{inj_h},p),\quad \forall\ p\in P,\quad s_{inj_c},\ S_{inJ_c},\quad s_{inj_h}\in S_{inJ_h} \qquad (2\text{-}18)$$

Constraints (2-19) and (2-20) ensure that the times at which units are active are synchronized when direct heat integration takes place. Finishing times for the tasks in the integrated units are the same. This constraint may be relaxed for operations requiring preheating or precooling and is dependent on the process.

$$tp(s_{inj_h}, p) \geq tp(s_{inj_c}, p) - H(1 - x(s_{inj_c}, s_{inj_h}, p)), \quad \forall\ p \in P,\ \ s_{inj_c}, S_{inJ_c},\ \ s_{inj_h} \in S_{inJ_h} \qquad (2\text{-}19)$$

$$tp(s_{inj_h}, p) \leq tp(s_{inj_c}, p) + H(1 - x(s_{inj_c}, s_{inj_h}, p)), \quad \forall\ p \in P,\ \ s_{inj_c}, S_{inJ_c},\ \ s_{inj_h} \in S_{inJ_h} \qquad (2\text{-}20)$$

Constraints (2-21) and (2-22) ensure that if indirect heat integration takes place, the time a unit is active will be equal to the time a heat storage unit starts either to transfer or receive heat.

$$tu(s_{inj}, p) \geq tu(u, p) - H(y(s_{inj}, p) - z(s_{inj}, u, p)), \quad \forall\ p \in P,\ \ u \in U,\ \ s_{inj} \in S_{inJ_c}, S_{inJ_h} \quad (2\text{-}21)$$

$$tu(s_{inj}, p) \leq tu(u, p) + H(y(s_{inj}, p) - z(s_{inj}, u, p)), \quad \forall\ p \in P,\ u \in U,\ s_{inj} \in S_{inJ_c}, S_{inJ_h} \qquad (2\text{-}22)$$

Constraints (2-23) and (2-24) state that the time when heat transfer to or from a heat storage unit is finished will coincide with the time the task transferring or receiving heat has finished processing.

$$tp(s_{inj}, p) \geq tp(u, p) - H(y(s_{inj}, p) - z(s_{inj}, u, p)), \quad \forall\ p \in P,\ u \in U,\ s_{inj} \in S_{inJ_c}, S_{inJ_h} \qquad (2\text{-}23)$$

$$tp(s_{inj}, p) \leq tp(u, p) + H(y(s_{inj}, p) - z(s_{inj}, u, p)), \quad \forall\ p \in P, u \in U, s_{inj} \in S_{inJ_c}, S_{inJ_h} \qquad (2\text{-}24)$$

Constraints (2-6) and (2-7) have trilinear terms resulting in a nonconvex mixed integer nonlinear programming (MINLP) formulation. The bilinearity resulting from the multiplication of a continuous variable with a binary variable may be handled effectively with the Glover transformation (Glover, 1975). This is an exact linearization technique and as such will not compromise the accuracy of the model. Using Constraints (2-25) and (2-26), Constraint (2-6) can be rewritten as Constraint (2-27).

$$T_f(u, p) - T_o(u, p) \leq \Gamma(s_{inj_h}, u, p) + M(1 - z(s_{inj_h}, u, p)), \quad \forall\ p \in P\ \ s_{inj_h} \in S_{inJ_h}\ \ u \in U \quad (2\text{-}25)$$

$$T_f(u, p) - T_o(u, p) \geq \Gamma(s_{inj_h}, u, p) - M(1 - z(s_{inj_h}, u, p)), \quad \forall\ p \in P\ \ s_{inj_h} \in S_{inJ_h}\ \ u \in U \qquad (2\text{-}26)$$

$$Q\left(s_{inj_h}, u, p\right) = W(u) c_p \left(\Gamma\left(s_{inj_h}, u, p\right)\right), \ \forall \ p \in P, \quad s_{inj_h} \in S_{inJ_h}, \quad u \in U \tag{2-27}$$

The heat storage capacity, $W(u)$, is also a continuous variable and is multiplied with the continuous Glover transformation variable. This results in another type of bilinearity, which results in a nonconvex model. A method to handle this is a Reformulation-Linearization technique (Sherali and Alameddine, 1992) as discussed by Quesada and Grossmann (1995). This is demonstrated for Constraint (2-27), resulting in Constraints (2-28) to (2-34).

With lower and upper heat storage capacity and temperature bounds known

$$W^L \le W(u) \le W^U \tag{2-28}$$

$$\Gamma^L\left(s_{inj_h}, u\right) \le \Gamma\left(s_{inj_h}, u, p\right) \le \Gamma^U\left(s_{inj_h}, u\right) \tag{2-29}$$

$$\Gamma^L\left(s_{inj_h}, u, p\right) = 0 \tag{2-30}$$

$$Q\left(s_{inj_h}, u, p\right) \ge c_p\left(W^L \Gamma\left(s_{inj_h}, u, p\right) + \Gamma^L\left(s_{inj_h}, u\right) W(u) - W^L \Gamma^L\left(s_{inj_h}, u\right)\right) \tag{2-31}$$

$$Q\left(s_{inj_h}, u, p\right) \ge c_p\left(W^U \Gamma\left(s_{inj_h}, u, p\right) + \Gamma^U\left(s_{inj_h}, u\right) W(u) - W^U \Gamma^U\left(s_{inj_h}, u\right)\right) \tag{2-32}$$

$$Q\left(s_{inj_h}, u, p\right) \le c_p\left(W^U \Gamma\left(s_{inj_h}, u, p\right) + \Gamma^L\left(s_{inj_h}, u\right) W(u) - W^U \Gamma^L\left(s_{inj_h}, u\right)\right) \tag{2-33}$$

$$Q\left(s_{inj_h}, u, p\right) \le c_p\left(W^L \Gamma\left(s_{inj_h}, u, p\right) + \Gamma^U\left(s_{inj_h}, u\right) W(u) - W^L \Gamma^U\left(s_{inj_h}, u\right)\right) \tag{2-34}$$

The full linearization procedure is carried out for each of the trilinear terms. The linearization procedure for Constraint (2-7) is presented from Constraint (2-35) to (2-42).

$$T_o(u, p) - T_f(u, p) \le \Gamma\left(s_{inj_c}, p\right) + M\left(1 - z\left(s_{inj_c}, u, p\right)\right), \ \forall \ p \in P, \quad s_{inj_c} \in S_{inJ_c}, \quad u \in U \tag{2-35}$$

$$T_o(u, p) - T_f(u, p) \ge \Gamma\left(s_{inj_c}, p\right) - M\left(1 - z\left(s_{inj_c}, u, p\right)\right), \ \forall \ p \in P, \quad s_{inj_c} \in S_{inJ_c}, \quad u \in U \tag{2-36}$$

$$\Gamma^L\left(s_{inj_c}, u\right) \le \Gamma\left(s_{inj_c}, u, p\right) \le \Gamma^U\left(s_{inj_c}, u\right) \tag{2-37}$$

$$\Gamma^L\left(s_{inj_c}, u\right) = 0 \tag{2-38}$$

$$Q\left(s_{inj_c}, u, p\right) \ge c_p\left(W^L \Gamma\left(s_{inj_c}, u, p\right) + \Gamma^L\left(s_{inj_c}, u\right) W(u) - W^L \Gamma^L\left(s_{injc}, u\right)\right) \tag{2-39}$$

$$Q(s_{injc},u,p) \geq c_p\left(W^U\Gamma(s_{injc},u,p)+\Gamma^U(s_{injc},u)W(u)-W^U\Gamma^U(s_{injc},u)\right) \quad (2\text{-}40)$$

$$Q(s_{injc},u,p) \leq c_p\left(W^U\Gamma(s_{injc},u,p)+\Gamma^L(s_{injc},u)W(u)-W^U\Gamma^L(s_{injc},u)\right) \quad (2\text{-}41)$$

$$Q(s_{injc},u,p) \leq c_p\left(W^L\Gamma(s_{injc},u,p)+\Gamma^U(s_{injc},u)W(u)-W^L\Gamma^U(s_{injc},u)\right) \quad (2\text{-}42)$$

Bounds on the heat storage capacity will be determined by the available space in the plant, as batch plants usually operate in limited space.

The objective function is defined in Constraint (2-43). The second and third terms are the costs for cooling water and steam, respectively.

$$\max \sum_{s} price\left(s^p\right)qs\left(s^p,p\right)-C_cQ_c-C_hQ_h \;,\; \forall p=P,\, s^p\in S^p \quad (2\text{-}43)$$

$$Q_c=\sum_{p}\sum_{s_{injc}} cw\left(s_{injc},p\right),\; \forall p\in P \;,\; s_{injc}\in S_{inJc} \quad (2\text{-}44)$$

$$Q_h=\sum_{p}\sum_{s_{injh}} st\left(s_{injh},p\right),\; \forall p\in P \;,\; s_{inj_h}\in S_{inJ_h} \quad (2\text{-}45)$$

2.4 Case studies

In order to demonstrate the application of the proposed model, three literature case studies are now discussed.

Case I

This case study has been studied extensively in published literature (Papageorgio et al., 1994; Majozi, 2006; Chen and Chang, 2009). It is a simple batch plant which produces one product using three unit operations (reaction, filtration and distillation). From Figure 2.2, the reaction task requires two raw materials, raw A and raw B and gives heat during the course of the reaction. The intermediate product from the reaction is filtered to separate it from waste. Here it is assumed that the waste material is insignificant compared to the intermediate product from the filtration. The distillation task separates the intermediate product from filtration into product 1 and product 2 by using external heat in the reboiler.

The processing time, the maximum capacity of units, available storage and initial inventory for each state is given in Table 2.1. This plant exhibits an opportunity for energy saving by integration between the reaction task and distillation column if sufficient time for the production is chosen. From thermodynamic and experimental estimation the amount of utility is summarised in Table 2.2. A time horizon of 48 h was chosen to maximize the revenue.

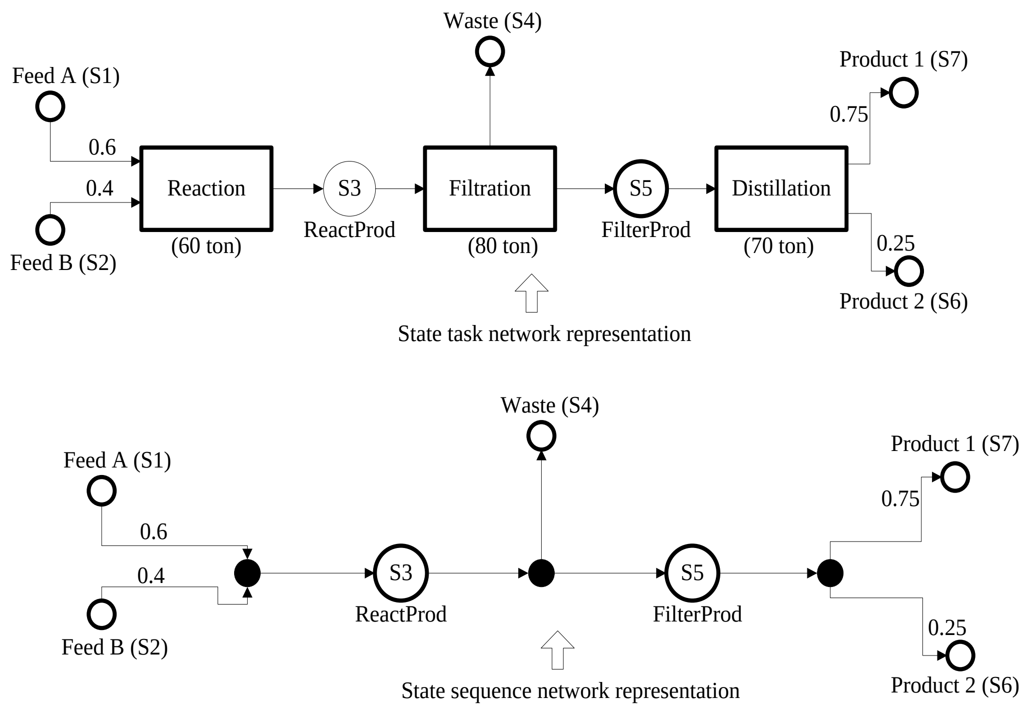


Figure 2.2. State task network (STN) and state sequence network (SSN) representation for Case I.

Table 2.1. Scheduling data for Case I.

Unit	Maximum capacity of unit (ton)	State	Initial inventory	Storage capacity	Selling price ($/ton)
Reactor	60	S1	UL	UL	0
		S2	UL	UL	0
Filter	80	S3	0	100	0
		S5	0	100	
Distiller	70	S6	0	UL	5
		S7	0	UL	5

UL = unlimited

Table 2.2. Utility requirements for Case I.

Task	Utility requirement	Operation	Amount (ton/h)	Duration of task (h)	Utility cost ($/ton)
Reaction	Cooling water	Standalone	$(1.59 + 0.10\,mu(s_{inj}, p))/h$	2	4
	Cooling water	Heat integrated	$(1.0 + 0.060\,mu(s_{inj}, p))/h$	3	4
Filtration	None	Standalone	$0\ t/h$	1	
Distillation	Steam	Standalone	$(0.044 + 0.0035\,mu(s_{inj}, p))/h$	2	200
	Steam	Heat integrated	$(0.02 + 0.0016\,mu(s_{inj}, p))/h$	2	200

Results and discussion

The proposed model for this case study was solved on a 1.8 GHz Pentium 4 personal computer using GAMS 22.0 with CPLEX as the MILP solver for fair comparison. The literature models were solved on a 2.0 GHz Pentium 4 using GAMS 22.3 with CPLEX as the MILP solver. The results for the literature models were taken directly from the paper of Chen and Chang (2009). The computational results for this case study are presented in Table 2.3. For a standalone operation, the proposed model and the model of Chen and Chang (2009) perform almost equally well and better than the model of Majozi (2006). In the heat integrated mode, all the models get the same optimal objective value of 3644.6, which is an improvement of 18% from the standalone scenario. The proposed model gave the lowest CPU time compared to the other literature models (34 s for the proposed model vs 9812.7 s for Majozi (2006) and 272 s for Chen and Chang (2009)). Chen and Chang (2009) reduce the computational time for this problem by task decomposition which is problem specific and applied only for this case study. The amount of material processed by a task and the starting and finishing times of tasks are shown in Figure 2.3.

Table 2.3. Computational results for Case I.

	Standalone				Heat-integrated				
	Papageorgiou et al. (1994)	Majozi (2006)	Chen & Chang (2009)	Proposed	Papageorgiou et al. (1994)	Majozi (2006)	Chen & Chang (2009)	Chen & Chang (2009) (Decomp.)	Proposed
Binary variables	142	72	72	118	188	96	244	186	138
Objective value	2944.1	3081.8	3081.8	3081.8	3644.6	3644.6	3644.6	3644.6	3644.6
Product 1 (t)	945	990	990	990	720	720	720	720	720
Product 2 (t)	315	330	330	330	240	240	240	240	240
Steam (t)	10.4	10.9	10.9	10.9	3.6	3.6	3.6	3.6	3.6
Cooling water (t)	318.8	334	334	334	107.2	107.2	107.2	107.2	107.2
Margin of Optimality (%)	5	0	0	0	5	< 1.3	< 5.36	0	0
Integrality gap (%)	4.76	0	0	0	0.37	0	0	0	0
CPU time (s)	N/A	24.5	0.375	0.203	N/A	9812.7	272	0.859	37

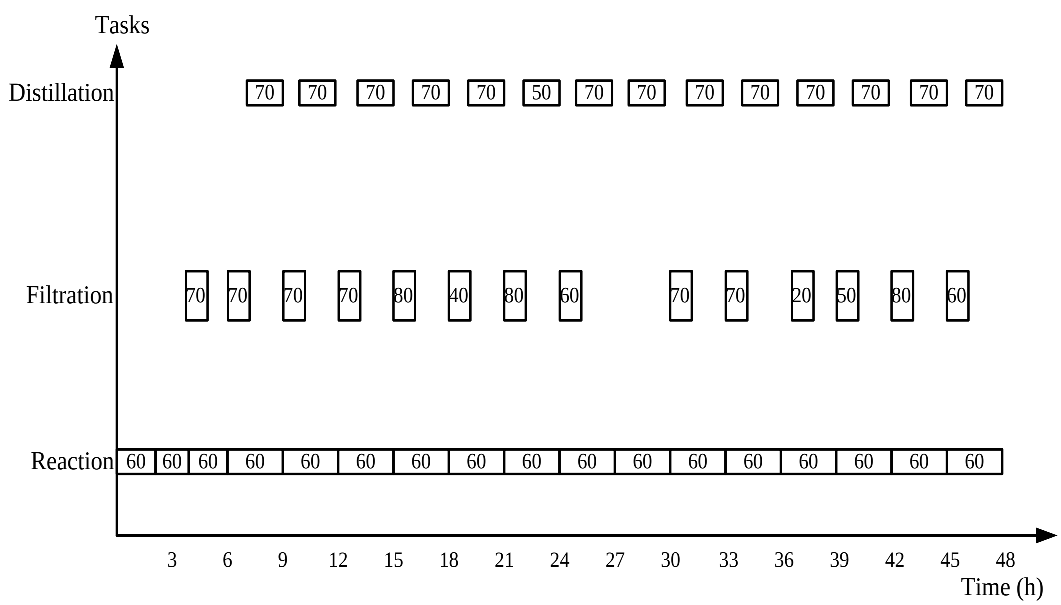
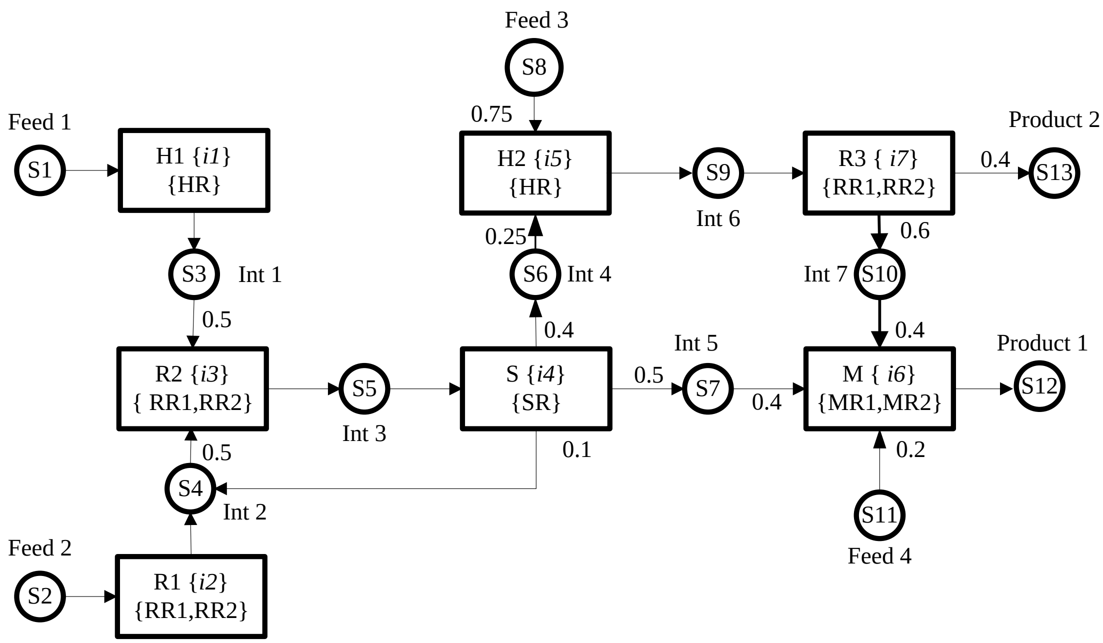


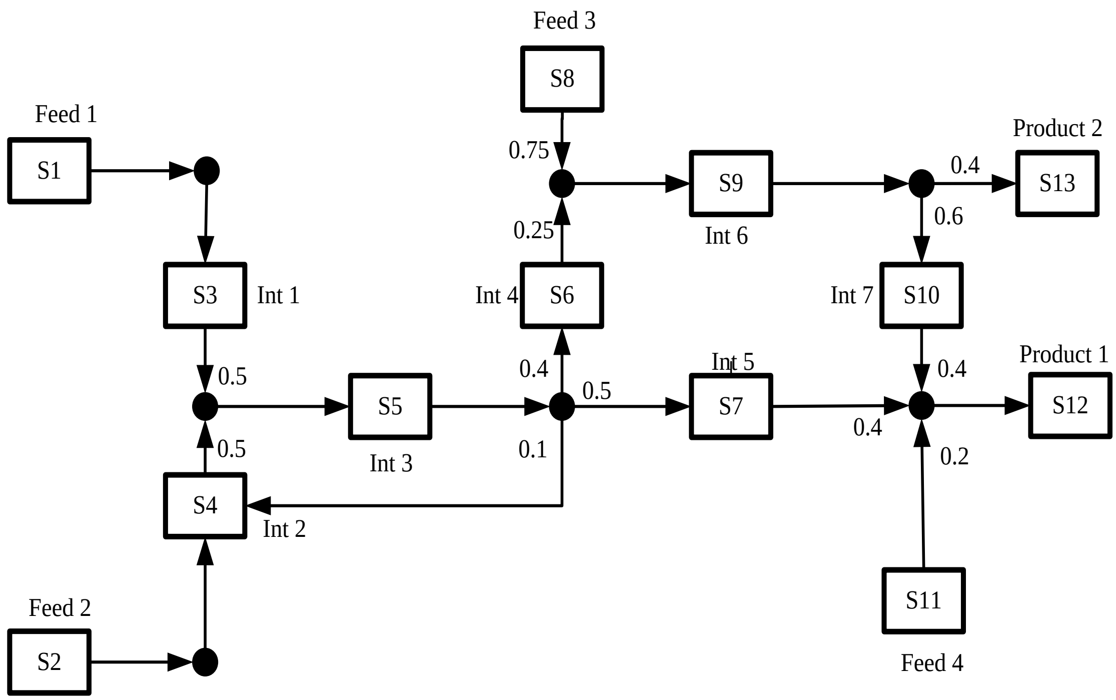
Figure 2.3. Gantt chart for Case I, heat integrated operation.

Case II

This example, which was first examined by Sundaramoorthy and Karimi (2005), is studied extensively in literature. It is a relatively complex problem and is often used in literature to check the efficiency of models in terms of optimal objective value and CPU time required. The plant has many common features of a multipurpose batch plant, with the following features: units performing multiple tasks, multiple units suitable for a task, states shared by multiple tasks and different products produced following different production paths. The state task and state sequence representations for this case study are depicted in Figure 2.4. The scheduling data are modified to incorporate heat integration opportunities and are presented in Tables 2.4 and 2.5. Data necessary for heat integration are presented in Table 2.6.



(a)



(b)

Figure 2.4 (a) STN representation and (b) SSN representation for Case II.

Table 2.4. Scheduling data for Case II.

Unit	Capacity	Suitability	Mean processing time (h)
Heater	100	H1, H2	1, 1.5
Reactor 1	100	RX1, RX2, RX3	2, 1, 2
Reactor 2	150	RX1, RX2, RX3	2, 1, 2
Separator	300	Separation	3
Mixer 1	200	Mixing	2
Mixer 2	200	Mixing	2

Table 2.5. Scheduling data for Case II.

State	Description	Storage capacity (ton)	Initial amount (ton)	Revenue (cu/ton)
S1	Feed 1	unlimited	unlimited	0
S2	Feed 2	unlimited	unlimited	0
S3	Intermediate 1	100	0	0
S4	Intermediate 2	100	0	0
S5	Intermediate 3	300	0	0
S6	Intermediate 4	150	50	0
S7	Intermediate 5	150	50	0
S8	Feed 3	unlimited	unlimited	0
S9	Intermediate 6	150	0	0
S10	Intermediate 7	150	0	0
S11	Feed 4	unlimited	unlimited	0
S12	Product 1	unlimited	0	1000
S13	Product 2	unlimited	0	1000

Parameters	Values
Specific heat capacity for heat storage c_p (kJ/kg°C)	4.2
Steam cost (cu/kWh)	10
Cooling water cost (cu/kWh)	2
$\Delta T^{\min}$ (°C)	10
T^L (°C)	20
T^U (°C)	180
W^L (ton)	1
W^U (ton)	3

Table 2.6. Heating/cooling requirements for Case II.

Reaction	Type	Heating/cooling requirement (kWh)	Operating temperature (ºC)
		Fixed batch size	
RX1	exothermic	60 (cooling)	100
RX2	endothermic	80 (heating)	60
RX3	exothermic	70 (cooling)	140
		Variable batch size	
RX1	exothermic	$5+0.25mu(s_{inj},p)$ cooling	100
RX2	endothermic	$20+0.6mu(s_{inj},p)$ heating	60
RX3	exothermic	$5+0.3mu(s_{inj},p)$ cooling	140

Results and discussion

In order to compare the results, the same computer as that used by Stamp and Majozi (2011), a Pentium 4 with 3.2 GHz processor and 512MB RAM was used. CPLEX and CONOPT 2 and GAMS 22.0 were used to solve the MILP and nonlinear programming problem (NLP) problems, respectively. DICOPT 2 was used as the interface for solving the MINLP problem. The computational results for this case study are presented in Table 2.7.

The proposed model and the model of Stamp and Majozi (2011) achieve the same optimal objective value of 222 000 for standalone operation in a similar CPU time. For direct heat integration, the proposed model gets an optimal objective value of 222 660, which looks inferior compared to the objective value of 222 840. This is because the model of Stamp and Majozi (2011) does not consider the offset time utility requirements if the heat integrated operations have different durations. It is better explained with the Gantt chart obtained for direct heat integration by the model of Stamp and Majozi (2011) in Figure 2.5. The model assumes there is no need for cooling water required by Reaction 3 in Reactor 1, since it is integrated with Reaction 2 in Reactor 2. However, as it can be seen in the Gantt chart, there must be some amount of cooling water for the duration of

Reaction 3 when it is not in the integration mode. The feasible Gantt chart for this scenario for the proposed model is given in Figure 2.6.

For indirect heat integration, using a time horizon of 10 h, both models give the same optimal objective value of 224 000 and the proposed model gives a better CPU time. It is interesting to note that by having heat storage, the utility requirement is reduced to 0.0 signifying that operating a heat integrated batch plant with heat storage is the best way to reduce the utility load.

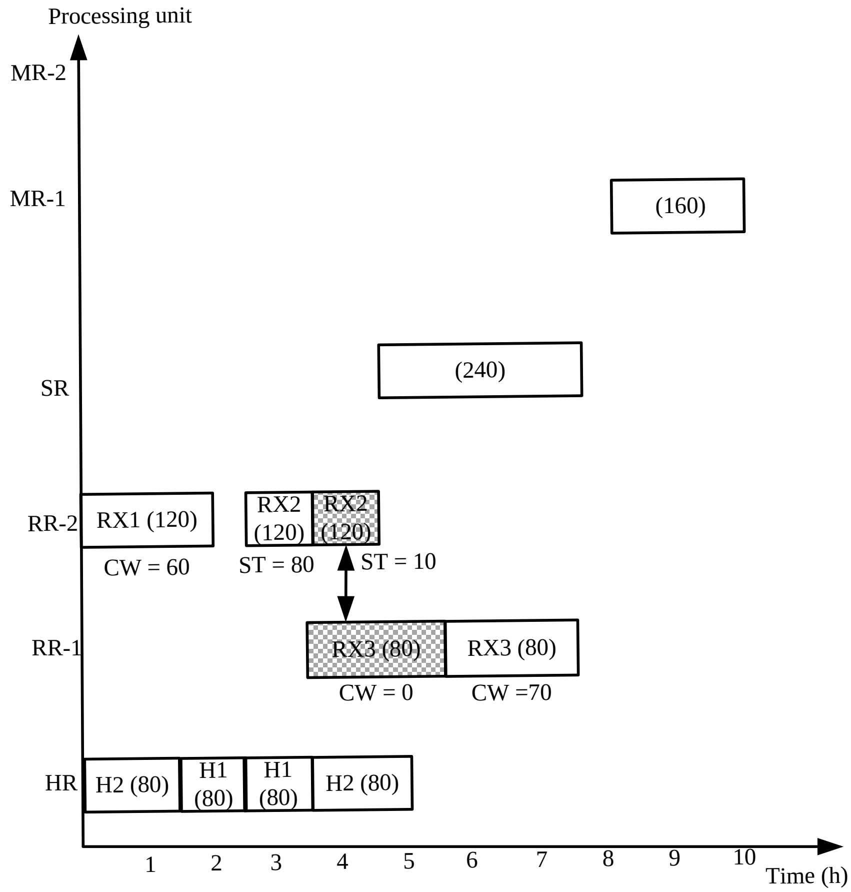


Figure 2.5. Gantt chart for direct heat integration using the model of Stamp and Majozi (2011), for Case II.

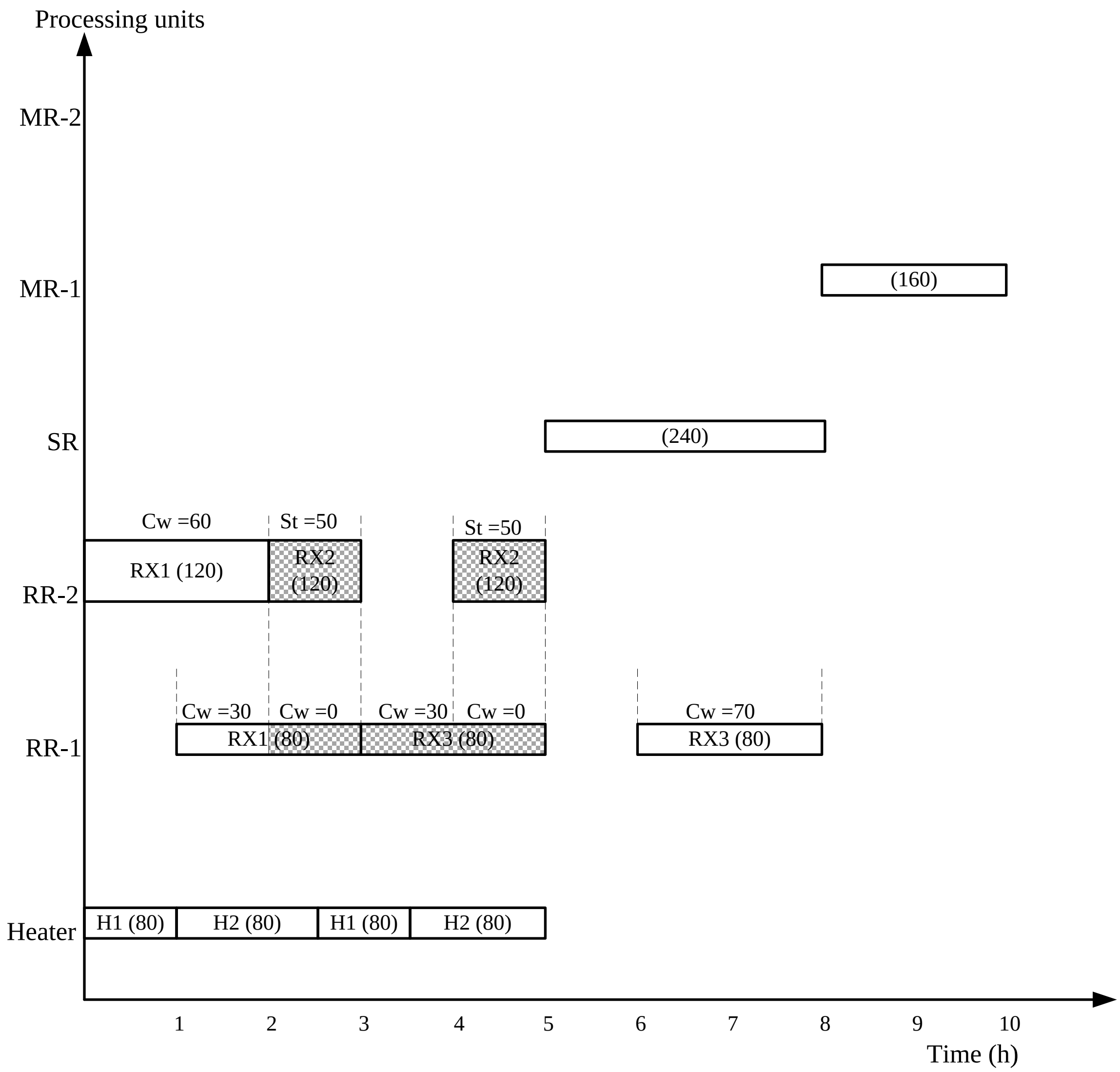


Figure 2.6. Gantt chart for direct integration for a time horizon of 10 h, proposed model.

For a time horizon of 12 h, the model by Stamp and Majozi (2011) required more than two days to solve, while the proposed model significantly reduced the required CPU time. (15.4 s for the proposed model vs. 238 640 s for Stamp and Majozi, 2011). A better optimal objective value of 287 858 was obtained by the proposed model as compared to 287 640 by Stamp and Majozi (2011). This indicates that the use of efficient scheduling techniques as a platform for a heat integration model improves the computational efficiency, both in terms of CPU time and optimal objective value for heat integration in multipurpose batch plants.

The amount of material processed, the starting and finishing times, the amount of heat exchanged between units and heat storage and the utility requirements of the units for a time horizon of 12 h is given in Figure 2.7. The shaded boxes indicate that the unit is integrated with storage. The model was extended for solving with variable batch sizes in this case study in order to avoid the limitations posed by fixing the batch size (Stamp and Majozi (2011)). Table 2.8 shows the results obtained for variable batch size using the proposed model. For 10 h and 12 h time horizons, the proposed model solved in a few seconds of CPU time. For variable batch size, for both time horizons, the objective values were significantly higher than for fixed batch size showing that it is better to operate the plant using variable batch sizes.

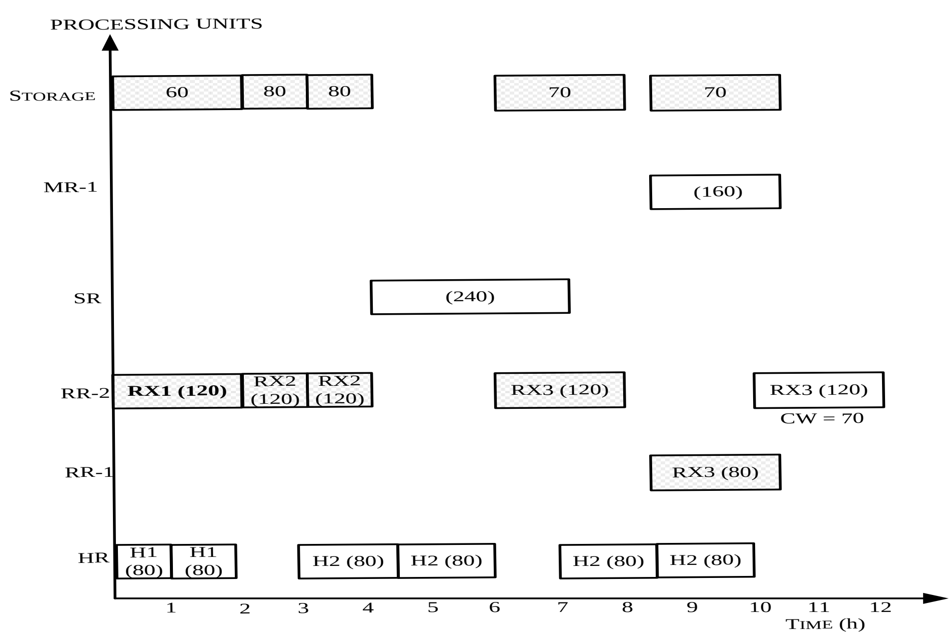


Figure 2.7 Gantt chart for indirect heat integration, proposed model.

Table 2.7. Computational results for Case II, fixed batch size with units operating at 80% capacity.

	Standalone operation, Stamp and Majozi (2011)	Standalone operation, Proposed	Direct heat integration, Stamp and Majozi (2011)	Direct heat integration, Proposed	Indirect heat integration, Stamp and Majozi (2011)	Indirect heat integration, Proposed
			H = 10			
Performance index (cost units)	222 000	222 000	222 840	222 660	224 000	224 000
External cold duty (kWh)	200	200	130	195	0	0
External hot duty (kWh)	160	160	90	95	0	0
Heat storage capacity (ton)					1.905	1.905
Initial heat storage temperature (^oC)					82.5	82.5
CPU time (s)	5.3	1	5.2	1.24	68	7.8
Binary variables	66	101	114	125	156	161
Time points	7	6	7	6	7	6
			H = 12			
Performance index (cost units)	285 860	285 860	286 540	286 100	287 640	287 858
External cold duty (kWh)	270	270	130	300	130	70
External hot duty (kWh)	160	160	120	130	10	0
Heat storage capacity (ton)					5	1.905
Initial heat storage temperature (^oC)					87.143	82.5
CPU time (s)	7.7	1.9	10.2	3.3	238 896	15.3
Binary variables	99	155	171	183	206	189
Time points	9	7	9	7	11	7

Table 2.8. Computational results for variable batch size, Case II.

	Standalone operation, Proposed	**Direct heat integration, Proposed**	**Indirect heat integration, Proposed**
H = 10			
Performance index (cost units)	418 450	418 999	419 998
External cold duty (kWh)	255	160	0
External hot duty (kWh)	104	69	0
Heat storage capacity (ton)			1.238
Initial heat storage temperature (^{o}C)			81.3
CPU time (s)	2	14	16.1
Binary variables	101	125	161
Time points	6	6	6
H = 12			
Performance index (cost units)	607 365	608 417	609 998
External cold duty (kWh)	337.5	300	0
External hot duty (kWh)	196	130	0
Heat storage capacity (ton)			1.143
Initial heat storage temperature (^{o}C)			75.694
CPU time (s)	1.9	26	124
Binary variables	119	147	189
Time points	7	7	7

Case III – Industrial Case Study

The STN for the process is shown in Figure. 2.8 (a) and the SSN is shown in Figure. 2.8 (b). The scheduling data may be obtained from Tables 2.9, 2.10 and 2.11. The plant consumes 55% of the steam utility in an agrochemical facility. Each of the units processes a fixed batch size of eight tons, 80% of design capacity. The process requires three consecutive chemical reactions, which take place in four available reactors. Reaction 1 takes place in either Reactor 1 or Reactor 2 and takes two hours. The intermediate from Reaction 1 is then transferred either to Reactor 3 or Reactor 4, where two consecutive reactions take place. Reaction 2 takes three hours and Reaction 3, one hour. Reaction 2 is highly exothermic. For operational purposes, these two consecutive reactions take place in a single reactor. Both the second and third reactions form sodium chloride as a byproduct. The intermediate from Reaction 3 is transferred to one of three Settlers, to separate the sodium chloride from the aqueous solution containing the active ingredient. This process takes one hour. This salt-free solution is then transferred to one of two Evaporators, where steam is used to remove excess water from the product, which takes three hours. This water is dispensed with as effluent. The final product is collected in storage tanks before final formulation, packaging and transportation to customers. This is an example of a sequential, mainly multiproduct process. The temperatures for the exothermic second reaction (150°C) and endothermic evaporation stage (90°C) allow for possible heat integration. Necessary heat integration data for the industrial case study may be found in Table 2.12, with heating and cooling requirements summarised in Table 2.13.

Table 2.9. Scheduling data for industrial case study.

State	Storage capacity (ton)	Initial amount (ton)	Revenue (cu/ton)
s1	Unlimited	unlimited	0
s2	Unlimited	0	0
s3	100	0	0
s4	100	0	0
s5	300	0	0
s6	150	0	100
s7	150	0	0
s8	Unlimited	0	0
s9	unlimited	unlimited	0
s10	unlimited	unlimited	0
s11	Unlimited	unlimited	0

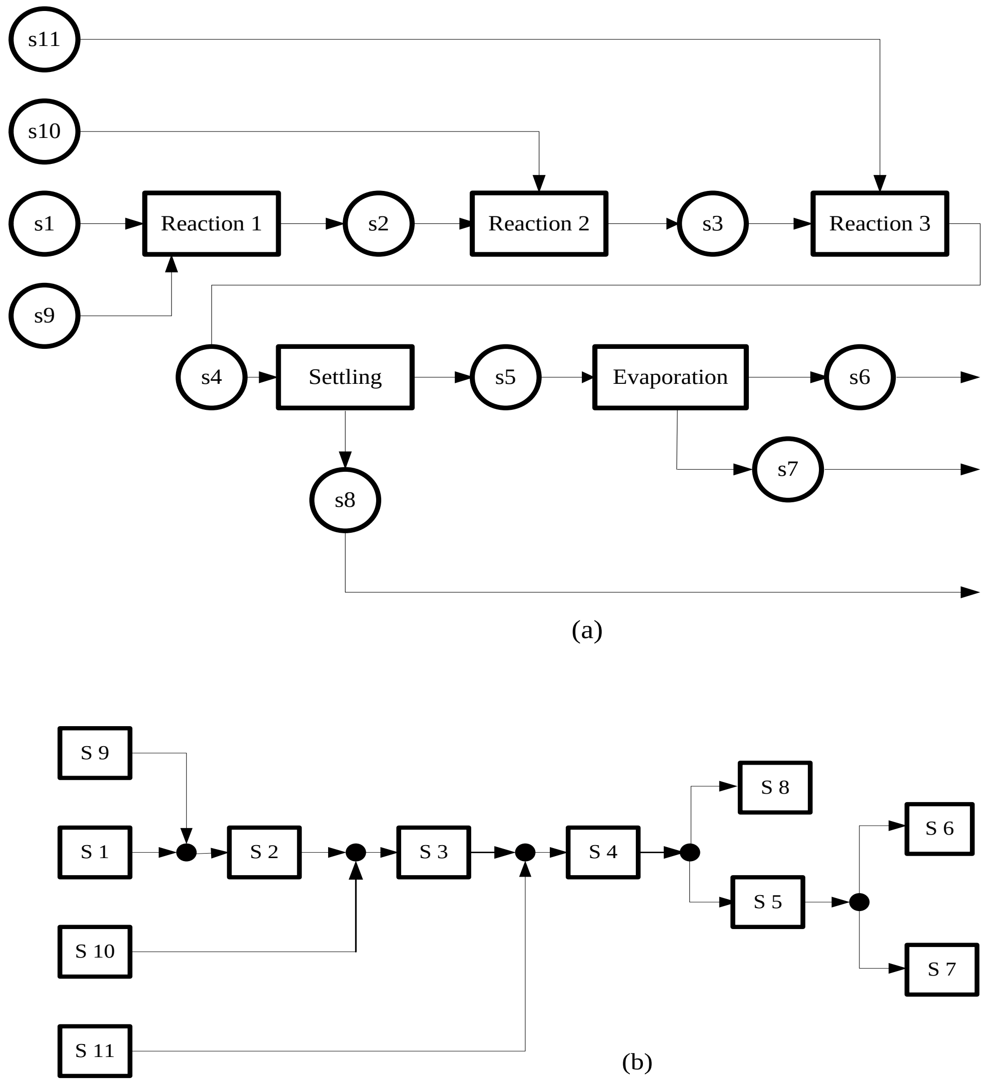


Figure 2.8. STN (a) and SSN (b) representation for Case III.

Table 2.10. Scheduling data for industrial case study.

Unit	Capacity	Suitability	Mean processing time (h)
R1	10	RX1	2
R2	10	RX1	2
R3	10	RX2, RX3	3, 1
R4	10	RX2, RX3	3, 1
SE1	10	Settling	1
SE2	10	Settling	1
SE3	10	Settling	1
EV1	10	Evaporation	3
EV2	10	Evaporation	3

Table 2.11. Stoichiometric data for industrial case study.

State	Ton/ton output	Ton/ton product
s1	0.20	
s9	0.25	
s10	0.35	
s11	0.20	
s7		0.7
s8		1

Table 2.12. Heat integration data for industrial case study.

Parameter	Value
Specific heat capacity, c_p (kJ/kg°C)	4.2
Product selling price (cu/ton)	100
Steam cost (cu/ton)	15
Cooling water cost (cu/ton)	8
$\Delta T^{\min}$ (°C)	5
W^L (ton)	0.2
W^U (ton)	1

Table 2.13. Heating/cooling requirements for industrial case study.

Reaction	Type	Heating/Cooling Requirement (ton)	Operating Temperature (ºC)
RX2	exothermic	5 (cooling)	150
Evaporation	endothermic	4 (heating)	90

Results and discussion

The computational results for the different models based on a continuous time framework are shown in Table 2.14. The computational results for the literature models are taken from Chen and Chang (2009). The case study is solved for a time horizon of 15 h for a fixed batch size, 80% of the maximum capacity of the units. The result for the proposed model for this case study was solved on a 1.8 GHz Pentium 4 personal computer using GAMS 22.0 with CPLEX as the MILP solver, for fair comparison. The literature models were solved on a 2.0 GHz Pentium 4 using GAMS 22.3 with CPLEX as the MILP solver.

For both cases, direct and indirect heat integration, the proposed model gave a better CPU time compared to the other literature models. For the heat integrated model an optimal objective value of 1267 was obtained which is an improvement of 18.3% compared to standalone operation, for which an optimal objective value of 1071 was obtained. The steam requirement was reduced from 12 ton in standalone operation to 0 ton and the cooling water decreased from 20 ton to 18 ton. The proposed model was extended to solve for indirect heat integration and gave an optimal objective value of 1355.8, which is 26.6% better when compared to the standalone operation. The use of heat storage as a medium to integrate the different batch processes if the processes are not performed at the same time will help in reducing utility requirement. The Gantt chart for a time horizon of 15 h for indirect heat integration using the proposed model is depicted in Figure 2.9. The bold boxes represent heat integration between processing units and the shaded boxes represent heat integration between processing units and heat storage.

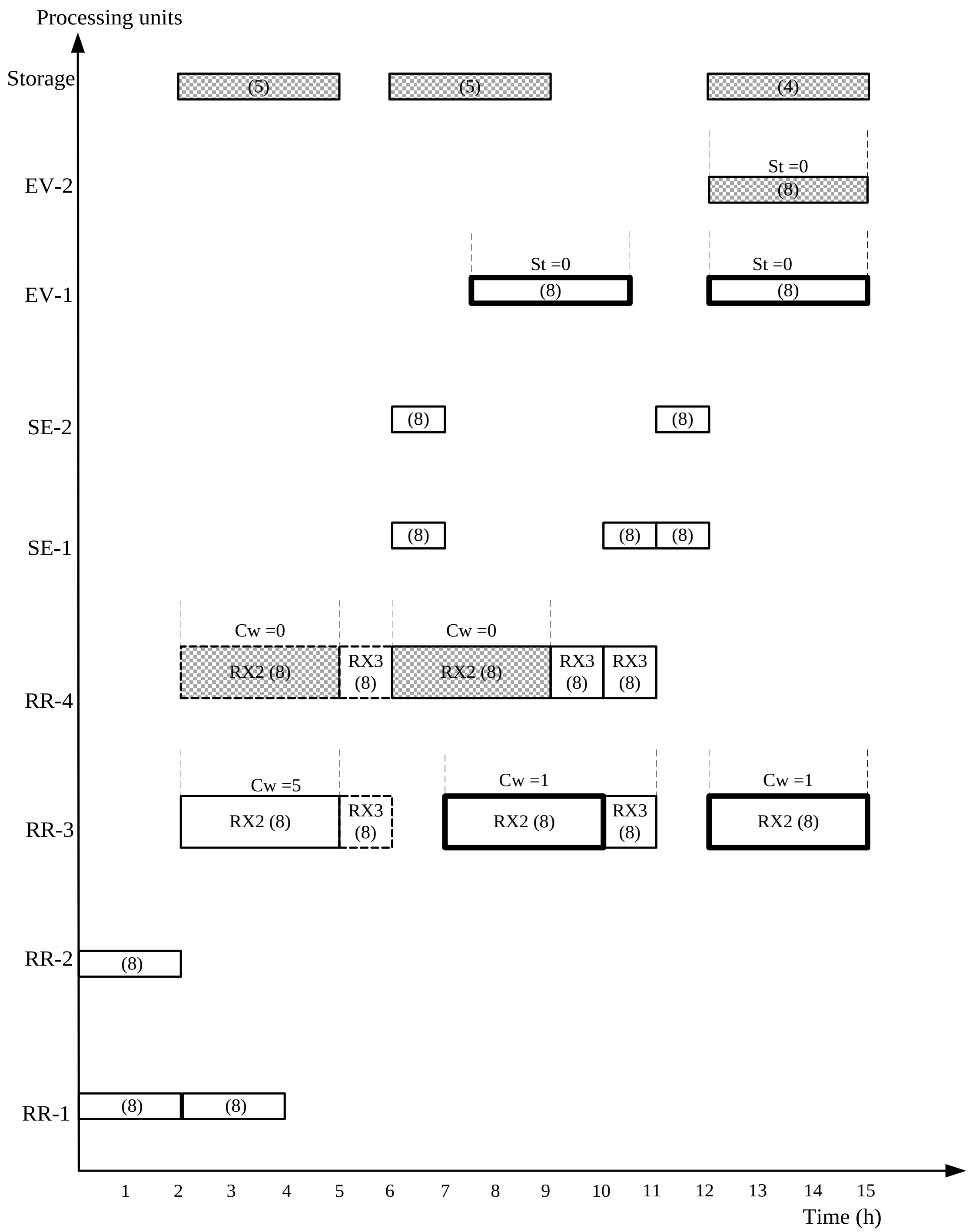


Figure 2.9 Gantt chart using indirect heat integration for Case III, propose model.

Table 2.14. Computational results for fixed batch size, Case III.

	Standalone operation, Majozi (2006)	Standalone operation, Chen and Chang (2009)	Standalone, Proposed	Direct heat integration, Majozi (2006)	Direct heat integration, Chen and Chang (2009)	Direct heat integration, proposed	Indirect heat integration, Proposed
			H = 16				
Performance index (cost units)	1060	1071	1071	1256	1267	1267	1355.76
External cold duty (kWh)	12	12	20	18	18	18	7
External hot duty (kWh)	20	20	12	0	0	0	0
CPU time (s)	26	64.67	16.7	26	113.8	0.54	267
Margin of optimality	0	0	0	0	0	0	0
Time points	11	11	8	11	11	8	8

2.5 Conclusions

An efficient continuous time mathematical model for direct and indirect heat integration is presented. Most heat integration models rely on a predefined schedule, which leads to suboptimal results. This work incorporates heat integration into the scheduling framework and the model is then capable of solving for both direct and indirect heat integration. By using a heat storage vessel, a considerable reduction in utility consumption is achieved. Application of the proposed formulation with indirect heat integration to an industrial case study showed a 26.6% improvement in objective function compared to standalone operation. Different case studies were used to test the performance of the proposed model and it was found that the model is computationally superior, both in terms of optimal objective value and CPU time.

Nomenclature

Sets

J	=	$\{ j \mid j$ is a processing unit$\}$
J_c	=	$\{ j_c \mid j_c$ is a processing unit which may conduct tasks requiring heating$\}$
J_h	=	$\{ j_h \mid j_h$ is a processing unit which may conduct tasks requiring cooling$\}$
P	=	$\{ p \mid p$ is a time point$\}$
S	=	$\{ s \mid s$ is any state$\}$
S^p	=	$\{ s^p \mid s^p$ is any state which is a product$\}$
S_{inJ}	=	$\{ s_{inj} \mid s_{inj}$ is any task$\}$
S_{inJ_c}	=	$\{ s_{inj_c} \mid s_{inj_c}$ is a task requires heating$\}$
S_{inJ_h}	=	$\{ s_{inj_h} \mid s_{inj_h}$ is a task requires cooling$\}$
U	=	$\{ u \mid u$ is a heat storage unit$\}$

Continuous variables

$cw(s_{inj_h}, p)$	=	external cooling required by unit j_h conducting the task corresponding to state s_{inj_h} at time point p
$Q(s_{inj}, u, p)$	=	heat exchanged with heat storage unit u at time point p
$st(s_{inj_c}, p)$	=	external heating required by unit j_c conducting the task corresponding to state s_{inj_c} at time point p
$T_0(u, p)$	=	initial temperature in heat storage unit u at time point p
$T_f(u, p)$	=	final temperature in heat storage unit u at time point p
$tu(u, p)$	=	time at which heat storage unit commences activity
$tp(u, p)$	=	time at which heat storage unit ends activity

$tu(s_{inj}, p)$ = time at which a task start in unit j

$tp(s_{inj}, p)$ = time at which a task finishes in unit j

$W(u)$ = capacity of heat storage unit u

$mu(s_{inj}, p)$ = amount of batch processed by a task in unit j

$qs(s, p)$ = amount of state s stored at time point p

$xx(s_{inj_h}, s_{inj_c}, p)$ = amount of heat exchange at between the cold and hot task at time point p.

Q_c = total amount of external cooling required during the scheduling horizon considered

Q_h = total amount of external heating required during the scheduling horizon considered

$\Gamma(s_{inj}, u, p)$ = Glover Transformation variable

Binary variables

$$x(s_{inj_c}, s_{inj_h}, p) = \begin{cases} 1 \leftarrow & \text{if unit } j_c \text{ conducting the task corresponding to state } s_{inj_c} \text{ is integrated with unit } j_h \text{ conducting the task corresponding to state } s_{inj_h} \text{ at time point } p \\ 0 \leftarrow & \text{otherwise} \end{cases}$$

$$y(s_{inj}, p) = \begin{cases} 1 \leftarrow & \text{if a task start in unit } j \text{ at time point } p \\ 0 \leftarrow & \text{otherwise} \end{cases}$$

$$z(s_{inj}, u, p) = \begin{cases} 1 \leftarrow & \text{if unit } j \text{ conducting the task corresponding to state } s_{inj} \text{ is integrated with storage unit } u \text{ at time point } p \\ 0 \leftarrow & \text{otherwise} \end{cases}$$

Parameters

c_p = specific heat capacity of heat storage fluid

$E(s_{inj})$ = constant coefficient of amount of heat required by or removed from unit j conducting the task corresponding to state s_{inj}

$\xi(s_{inj}, p)$ = variable coefficient of amount of heat required by or removed from unit j conducting the task corresponding to state s_{inj}

M = any large number

$T(s_{inj})$ = operating temperature for processing unit j conducting the task corresponding to state s_{inj}

T^L = lower bound for heat storage temperature

T^U = upper bound for heat storage temperature

$\Delta T^{\min}$ = minimum allowable thermal driving force

$\tau(s_{inj})$ = duration of the task corresponding to state s_{inj} conducted in unit j

W^L = lower bound for heat storage capacity

W^U = upper bound for heat storage capacity

$\Gamma^L(s_{inj}, u)$ = lower bound for Glover transformation variable

$\Gamma^U(s_{inj}, u)$ = upper bound for Glover transformation variable

C_c = cost unit for cooling water

C_h = cost unit for steam

H = time horizon of interest

$price(s^p)$ price of a product

References

Adonyi, R., Romero, J., Puigjaner, L., Friedler, F., 2003. Incorporating heat integration in batch process scheduling. *Appl. Therm. Eng.* 23, 1743–1762.

Atkins, M.J., Walmsley, M.R.W., Neale, J.R., 2010. The challenge of integrating non-continuous processes – milk powder plant case study.*J. Clean. Prod*. 18, 927–34.

Boyadjiev, C.H.R., Ivanov, B., Vaklieva-Bancheva, N., Pantelides, C.C., Shah N., 1996. Optimal energy integration in batch antibiotics manufacture. *Comp. Chem. Eng*. 20:S31–S36.

Chen, C.L., Chang, C.Y., 2009. A resource-task network approach for optimal short-term/periodic scheduling and heat integration in multipurpose batch plants. *Appl. Therm. Eng*. 29, 1195–1208.

Fernández, I., Renedo, C.J., Pérez, S.F., Ortiz, A., Mañana, M., 2012 .A review: Energy recovery in batch processes. *Renewable and Sustainable Energy Reviews*. 16:2260–2277.

Fritzson, A., Berntsson, T., 2006. Efficient energy use in a slaughter and meat processing plant – opportunities for process integration. *J. Food Eng.* 76, 594–604.

Georgiadis, M.C., Papageorgiou, L.G., 2001. Optimal scheduling of heat-integrated multipurpose plants under fouling conditions. *Appl. Therm. Eng*. 21, 1675–97.

Glover F., 1975. Improved linear integer programming formulations of nonlinear integer problems. *Man. Sci*. 22(4):455e60.

Halim, I., Srinivasan, R., 2009. Sequential methodology for scheduling of heat-integrated batch plants. *Ind. Eng. Chem. Res*. 48, 8551–65.

Knopf, F.C., Okos, MR., Reklaitis, G.V., 1982. Optimal design of batch/semicontinuous processes. *Ind. Eng. Chem. Process Des. Dev*. 21, 79–86.

Kondili, E., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations. I. MILP formulation. *Comput. Chem. Eng.* 17, 211-227.

Majozi, T., 2006. Heat integration of multipurpose batch plants using a continuous-time framework. *Appl. Therm. Eng.* 26:1369–77.

Majozi, T., 2009. Minimization of energy use in multipurpose batch plants using heat storage: an aspect of cleaner production. *J. Clean. Prod*. 17, 945–50.

Majozi, T., Zhu, X.X., 2001. A novel continuous-time MILP Formulation for multipurpose bach plants. *Ind. Eng. Chem. Res*. 40, 5935-5949.

Mignon, D., Hermia, J., 1993. Using batches for modeling and optimizing the brewhouses of an industrial brewery. *Comput. Chem. Eng.* 17 S51–S56.

Papageorgiou, L.G., Shah, N., Pantelides, C.C., 1994. Optimal scheduling of heat-integrated multipurpose plants. *Ind. Eng. Chem. Res*. 33, 3168–186.

Quesada I, Grossmann, I.E., 1995. Global optimization of bilinear process networks with multicomponent flows. *Comp. Chem. Eng*. 19(12):1219e42.

Rašković, P., Anastasovski, A., Markovska, LJ., Meško, V., 2010. Process integration in bioprocess industry: waste heat recovery in yeast and ethyl alcohol plant. *Energy*. 35, 704-717.

Sanmartí, E., Friedler, F., Puigjaner, L., 1998. Combinatorial technique for short term scheduling of multipurpose batch plants based on schedule-graph representation. *Comp Chem Eng*. 22, S847–S850.

Seid, R., Majozi, T., 2012. A robust mathematical formulation for multipurpose batch plants**.** *Chem. Eng. Sci*. 68, 36–53.

Sherali, H.D, Alameddine, A., 1992. A new reformulation-linearization technique for bilinear programming problems. *J. Glob. Optim.* 2(4):379e410.

Stamp, J., Majozi, T., 2011. Optimal heat storage design for heat integrated multipurpose batch plants. *Energy*. 36(8), 1-13.

Sundaramoorthy, A., Karimi, I.A., 2005. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chem. Eng. Sci*. 60, 2679-2702.

Tokos, H., Pintarič, ZN., Glavič, P., 2010. Energy saving opportunities in heat integrated plant retrofit. *Appl. Therm. Eng*. 30, 36–44.

Uhlenbruck, S., Vogel, R., Lucas, K., 2000. Heat integration of batch processes. *Chem. Eng. Technol*. 23, 226–229.

Vaklieva-Bancheva, N., Ivanov, B.B., Shah, N., Pantelides, C.C., 1996. Heat exchanger network design for multipurpose batch plants. *Comput. Chem. Eng*. 20, 989–1001.

Vasenlenak, J.A, Grossmann, I.E, Westerberg, A.W., 1986. Heat integration in batch processing. *Ind. Eng. Chem. Process Des. Dev*. 25, 357–66.

3

CHAPTER 3

Simultaneous Optimization of Energy and Water Use in Multipurpose Batch Plants

Abstract

Major contributors to the overall running costs of multipurpose batch plants such as breweries, dairies, biochemical plants and agrochemical facilities are water and heating and cooling utilities. Water is used for process equipment cleaning, when the equipment is switched from one task to another due to the inherent sharing of equipment by different tasks. Heating and cooling are also unavoidable aspects of many plant facilities. There are operations where heat is generated and others where heat is required. This presents an opportunity for energy integration.

Energy and water optimization problems in multipurpose batch plants have been treated as separate entities in literature. The batch production schedule resulting from each formulation does not guarantee that the plant is operated optimally. Consequently, it is necessary to develop a formulation that simultaneously minimizes wastewater and external heating and cooling utilities. This would result in a production schedule with minimum cost.

This work presents a formulation that minimizes both water and energy, while simultaneously optimizing the batch process schedule. The scheduling framework used in this study is based on the formulation by Seid and Majozi (2012). This formulation has shown significant reduction in computational time while improving the objective

value. From a case study it was found that through only applying water integration the total cost is reduced by 11.6%, by applying only energy integration the total cost is reduced by 29.1% and by applying both energy and water integration the total cost is reduced by 34.6%. This indicates that optimizing water and energy integration in the same scheduling framework will reduce the operating cost and significantly lessen the process environmental impact.

Keywords: Wastewater minimization, energy integration, heat storage, multipurpose batch plant

3.1 Introduction

In recent years, batch processes have been getting more attention due to their suitability for the production of small volume, high value added products. The flexibility of batch plants allows the production of different products within the same facility. Despite the advantage of batch plants being flexible, they also pose a challenging task to design, synthesize and operate, compared to their continuous counterparts. Considerable advancement has been made for scheduling of batch operations as compared to minimizing energy and wastewater. In the past, batch industries could tolerate high inefficiencies in energy and water consumption due to the high value of final products which outstripped the production costs. However, greater public awareness of the impact of industrial pollution, more stringent environmental regulations and escalating raw materials, energy, and waste treatment costs have now encouraged the introduction of measures for more sustainable operations (Halim and Srinivasan, 2011). Since scheduling, energy and wastewater minimization for multipurpose batch plants go hand in hand, published work in those areas are reviewed.

3.1.1 Scheduling of batch plants

Significant research has been done on developing mathematical models to improve batch plant efficiency. The substantial advancement in modern computers allows the possibility of handling large and more complex problems by using optimization techniques. Excellent reviews of current scheduling techniques based on different time representations and associated challenges have been conducted (Méndez et al., 2006; Floudas and Lin, 2004; Shaik, 2006). In the reviews, with regard to time representation, the models are classified as slot based, event based and precedence based (sequence-based). In the slot based models the time horizon is divided into ‘‘nonuniform unknown slots’’ and tasks start and finish in the same slot (Pinto and Grossmann, 1994; Lim and Karimi, 2003; Liu and Karimi, 2008). On the other hand, slot models exist that use nonuniform unknown slots where tasks are allowed to continue to the next slots (Schilling and Pantelides, 1996; Karimi and McDonald 1997; Reddy et al., 2004; Sundaramoorthy and Karimi, 2005; Erdirik-Dogan and Grossmann, 2008; Susarla et al., 2010). The event based models can also be categorized into those that use uniform unknown events, where the time associated with the events is common across all units, (Maravelias and Grossmann, 2003; Castro et al., 2004) and those that use unit specific events where the time associated with the events can be different across the units (Ierapetritou and Floudas, 1998; Majozi and Zhu, 2001; Janak and Floudas, 2008; Shaik et al., 2006; Shaik and Floudas, 2009; Li et al., 2010). The heterogeneous location of events across the units gives fewer event points as compared to both the global event based and slot based models. As a result, unit specific event based models are computationally superior. The sequence-based or precedence-based representation uses either direct precedence (Hui and Gupta, 2000; Liu and Karimi, 2007) or indirect precedence sequencing of pairs of tasks in units (Méndez et al., 2000, 2001; Méndez and Cerdá, 2003; Ferrer-Nadal et al., 2008). The models do not require pre-postulation of events and slots. Seid and Majozi (2012) presented a mixed integer linear programming (MILP) formulation based on the state sequence network and unit specific time points, which can handle proper sequencing of tasks and fixed intermediate storage (FIS) policy. The model results in a reduction of event or time points required and as a result, give better performance in terms of objective value and CPU time required when compared to previous literature models.

3.1.2 Energy integration in batch plants

A common feature of many batch plants is that they utilize fossil fuels as the energy source. The dependence of these batch plants on fossil fuels as the energy source results in increasing energy cost and negatively impacts on the environment. Improving energy use through process integration could be a solution to efficient use of energy. Energy savings in batch plants were neglected in the past because it was believed that they were not as large in magnitude as in continuous cases. A detailed literature review in this area was presented in Section 2.1 of Chapter 2.

3.1.3 Wastewater minimization in batch plants

Wastewater is generated in batch plants during cleaning of multipurpose equipment and when water is used as a solvent. Tight environmental regulations and increased public awareness demand that batch plants consider rational use of water during their operation. Many researchers have developed methodologies for the efficient use of water through direct reuse, indirect reuse and regeneration of wastewater. Direct reuse consists of recycle and reuse. Recycle refers to the reuse of an outlet wastewater stream from a processing unit in the same unit, while reuse refers to the use of an outlet wastewater stream from a processing unit in another processing unit. Indirect reuse is when wastewater is temporarily stored in a storage vessel and later reused in a processing unit requiring water. The optimization of the batch production schedule, water reuse and wastewater treatment in one single problem has been developed by Cheng and Chang, (2007). At the end of the optimization, the production schedule, the number and sizes of buffer tanks and the physical configuration of the pipeline network were obtained. The time horizon was also discretized in this work.

Majozi and Gouws (2009) addressed wastewater minimization problems with multiple contaminants. In this method, the batch production schedule and the minimum wastewater target were obtained simultaneously making this methodology more rigorous. However, wastewater regeneration was not considered. Adekola and Majozi (2011).

extended the work of Majozi and Gouws (2009) by incorporating wastewater regeneration for further improvement of water utilization.

From the review it can be seen that wastewater minimization and heat integration in batch plants are addressed separately. As far as the knowledge of the author is concerned single research work was presented by Halim and Srinivasan (2011) to address this literature gap. The overall problem is decomposed into three parts scheduling, heat integration and water reuse optimization and solved sequentially. Batch scheduling is solved first to meet an economic objective function. Next, alternate schedules are generated through a stochastic search based integer cut procedure. For each resulting schedule, minimum energy and water reuse targets are established and networks identified.

In this work a contribution is made to close the literature gap by simultaneously solving energy integration and wastewater minimization problems in the same scheduling framework. The rest of this chapter is organized as follows. Section 3.2 defines the problem statement. Section 3.3 describes the detailed mathematical formulation. Section 3.4 describes the application of the mathematical model to literature problems. Finally conclusions are drawn from this work.

3.2 Problem statement

Given

i) The production recipe (STN or SSN representation)
ii) The capacity of units and the type of tasks each unit can perform
iii) The maximum storage capacity for each material
iv) The task processing times
v) Hot duties for tasks requiring heating and cold duties for tasks that require cooling
vi) Operating temperatures of heat sources and heat sinks
vii) Minimum allowable temperature differences
viii) The material's heat capacity
ix) The units' washing times

x) The mass load of each contaminant

xi) The concentration limits of each contaminant

xii) The costs of raw materials, products and utilities

xiii) The scheduling horizon (for profit maximization problem)

xiv) Production demand (for makespan minimization problem)

Determine:

a) The production schedule (i.e. allocation of tasks to units, timing of all tasks and batch sizes),

b) The heat exchange configuration and

c) The direct water-reuse network.

3.3 Mathematical formulation

The SSN recipe representation and an uneven discretisation of the time horizon were used. The scheduling model by Seid and Majozi (2012) was adopted since it has proven to result in fewer binary variables, CPU time and a better optimal objective value compared to other scheduling models.

3.3.1 Heat integration model

The mathematical model is based on the superstructure in Figure 3.1. Each task may operate using either direct or standalone mode by using only external utilities. If direct integration is not sufficient to satisfy the required duty, external utilities may makeup for any deficit. The model in this chapter is different from the one presented in Chapter 2 because it caters for the temperature change in a processing task during its operation whereas the model in the previous chapter only works for a task operated with a constant or very small variation in temperature.

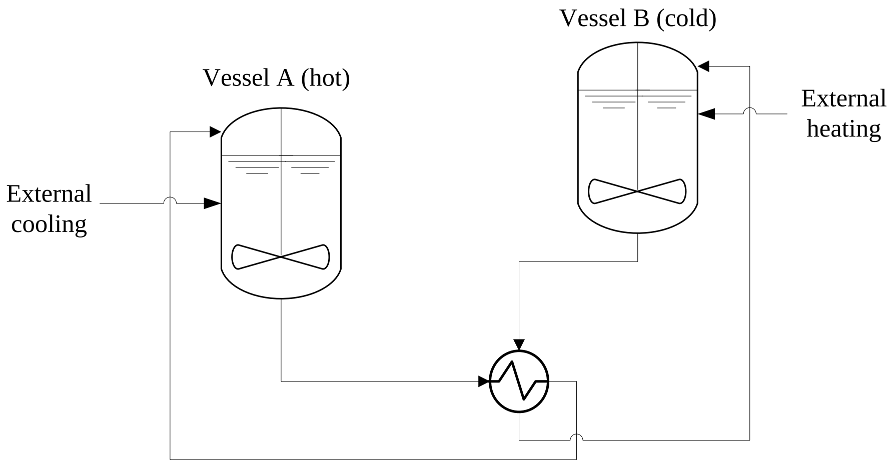


Figure 3.1. Superstructure for the energy integration.

Constraints (3-1) and (3-2) are active simultaneously and ensure that one hot unit will be integrated with one cold unit when direct heat integration takes place, in order to simplify operation of the process. It is also possible for one unit to integrate with more than one unit at a given time point when ignoring the summation notation. Also, if two units are to be heat integrated at a given time point, they must both be active at that time point.

$$\sum_{s_{injc}} x(s_{inj_c}, s_{inj_h}, p, pp) \leq y(s_{inj_h}, p), \forall \; p, pp \in P, \quad s_{inj_h} \in S_{inJ_h}, \; s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}1)$$

$$\sum_{s_{injh}} x(s_{inj_c}, s_{inj_h}, p, pp) \leq y(s_{inj_c}, p), \forall \; p, pp \in P, \quad s_{inj_h} \in S_{inJ_h}, \; s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}2)$$

For better understanding, the difference between time point *p* and extended time point *pp* is explained using Figure 3.2. If a unit *j* that is active at time point *p* is integrated with more than one unit in different temperature and time intervals, an extended time point *pp* must be defined Unit *j1* active at time point *p* can be integrated with units *j2* and *j3* in different time and temperature intervals. At the beginning, unit *j1* is integrated with unit *j2* at time point *p* and the extended time point *pp* is the same as time point *p*. Later *j1* is integrated with unit *j3* in another time interval where extended time point *pp* equals to *p+1*. *pp* is equal to or greater than time point *p* and less than or equal to *n+p*, where *n* is a parameter which is greater than or equal to zero. If *n* equals 2 then a unit that is active at time point p can be integrated in three different intervals. The model should be solved starting from n equals zero and adding one at a time until no better objective value is achieved.

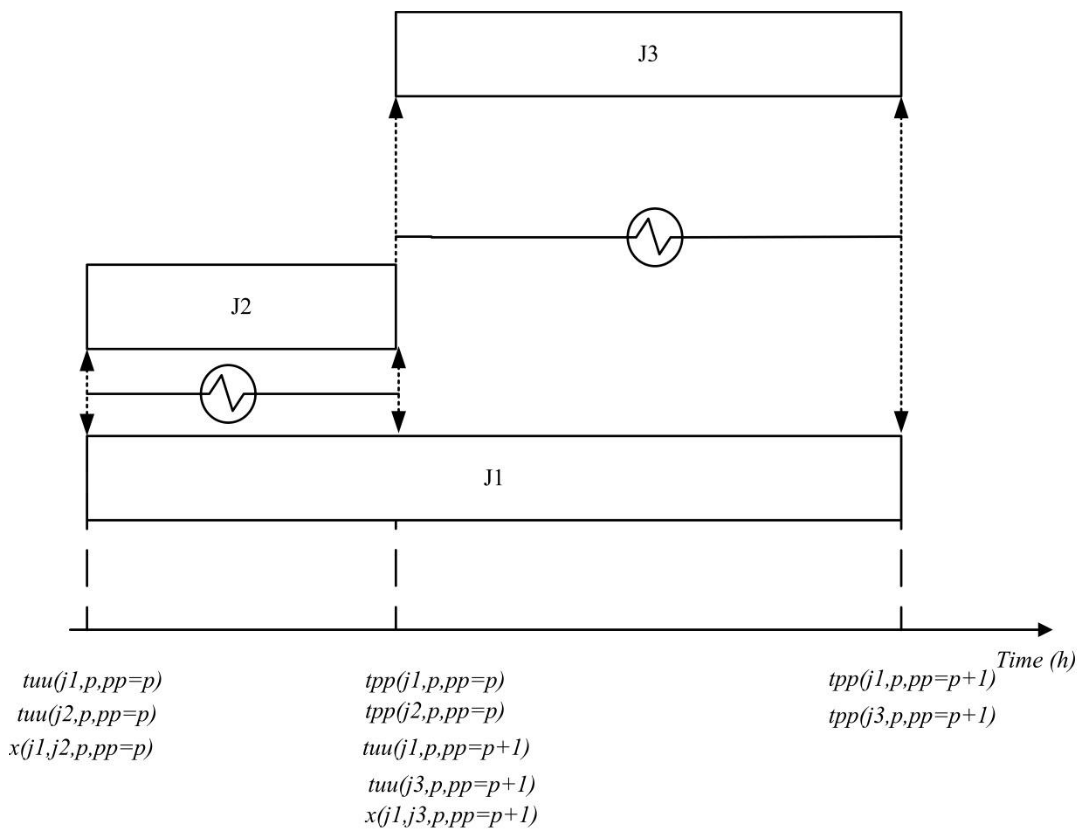


Figure 3.2. Differentiating time point *p* and extended time point *pp*.

Constraint (3-3) describes the amount of cooling load required by the hot unit from its initial temperature to its target temperature. In the event where the temperature in the reactor unit is fixed during exothermic reaction, the heat load becomes the product of the amount of mass that undergoes reaction and the heat of reaction.

$$cl(s_{inj_h}, p) = mu(s_{inj_h}, p)cp(s_{inj_h})\left(T^{in}_{s_{inj_h}} - T^{out}_{s_{inj_h}}\right), \ \forall\ p \in P,\ s_{inj_h} \in S_{inJ_h} \qquad (3\text{-}3)$$

Constraint (3-4) describes the amount of heating load required by the cold unit from its initial temperature to its target temperature. In the event where the temperature in the reactor unit is fixed during endothermic reaction, the heat load becomes the product of the amount of mass that undergoes reaction and the heat of reaction.

$$hl(s_{injc}, p) = mu(s_{injc}, p)cp(s_{injc})(T^{out}_{s_{injc}} - T^{in}_{s_{injc}}), \forall\ p \in P, \quad s_{injc} \in S_{inJc} \tag{3-4}$$

Constraints (3-5) and (3-6) describe the average heat flow for the hot and cold unit during the processing time respectively. This is the same as time average (TAM) model to address the energy balance during heat integration properly.

$$cl(s_{inj_h}, p) = avcl(s_{inj_h}, p)(tp(s_{inj_h}, p) - tu(s_{inj_h}, p)), \forall\ p \in P,\ s_{inj_h} \in S_{inJ_h} \tag{3-5}$$

$$hl(s_{inj_c}, p) = avhl(s_{inj_c}, p)(tp(s_{inj_c}, p) - tu(s_{inj_c}, p)), \forall\ p \in P,\ s_{inj_c} \in S_{inJ_c} \tag{3-6}$$

Constraints (3-7) and (3-8) define the heat load at time point *p* and extended time point *pp* for the cold and hot unit respectively.

$$hlp(s_{inj_c}, p, pp) = avhl(s_{inj_c}, p)(tpp(s_{inj_c}, p, pp) - tuu(s_{inj_c}, p, pp)), \forall\ p, pp \in P,\ s_{inj_c} \in S_{inJ_c} \tag{3-7}$$

$$clp(s_{injh}, p, pp) = avcl(s_{injh}, p)(tpp(s_{injh}, p, pp) - tuu(s_{injh}, p, pp)), \forall\ p, pp \in P,\ s_{injh} \in S_{inJh} \tag{3-8}$$

Constraints (3-9) and (3-10) are used to calculate the temperature of the hot and cold unit at the intervals.

$$clp(s_{inj_h}, p, pp) = mu(s_{inj_h}, p)cp(s_{inj_h})(T^{in}(s_{inj_h}, p, pp) - T^{out}(s_{inj_h}, p, pp)), \forall\ p, pp \in P,\ s_{inj_h} \in S_{inJ_h} \tag{3-9}$$

$$hlp(s_{injc}, p, pp) = mu(s_{inj_c}, p)cp(s_{inj_c})(T^{in}(s_{inj_c}, p, pp) - T^{out}(s_{inj_c}, p, pp)), \forall\ p, pp \in P,\ s_{inj_c} \in S_{inJ_c} \tag{3-10}$$

Constraint (3-11) states that the amount of heat exchanged by the hot unit with the cold units should be less than the cooling load required during the interval.

$$\sum_{s_{injc}} Ql(s_{inj_h}, s_{inj_c}, p, pp) \le clp(s_{inj_h}, p, pp), \forall\ p, pp \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c} \tag{3-11}$$

Constraint (3-12) states that the amount of heat exchanged by the cold unit with the hot units should be less than the heat load required during the interval.

$$\sum_{s_{injh}} Ql(s_{inj_h}, s_{inj_c}, p, pp) \le hlp(s_{inj_c}, p, pp), \forall\ p, pp \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c} \tag{3-12}$$

Constraints (3-13) and (3-14) state that the temperature of a unit at the start of an interval should be equal to the temperature at the end of the previous interval.

$$T^{in}\left(s_{inj_h}, p, pp\right) = T^{out}\left(s_{inj_h}, p, pp-1\right), \forall\, p, pp \in P,\ s_{inj_h} \in S_{inJ_h} \quad (3\text{-}13)$$

$$T^{in}\left(s_{inj_c}, p, pp\right) = T^{out}\left(s_{inj_c}, p, pp-1\right), \forall\, p, pp \in P,\ s_{inj_c} \in S_{inJ_c} \quad (3\text{-}14)$$

Constraints (3-15) and (3-16) state that the temperature at the start of the first interval, which is time point *p*, which is also *pp*, should be equal to the initial temperature of the task.

$$T^{in}\left(s_{inj_h}, p, pp\right) = T^{in}\left(s_{inj_h}\right), \forall\, p = pp,\ p, pp \in P,\ s_{inj_h} \in S_{inJ_h} \quad (3\text{-}15)$$

$$T^{in}\left(s_{inj_c}, p, pp\right) = T^{in}\left(s_{inj_c}\right), \forall\, p = pp,\ p = pp \in P,\ s_{inj_c} \in S_{inJc} \quad (3\text{-}16)$$

Constraints (3-17) and (3-18) ensure that the minimum thermal driving forces are obeyed when there is direct heat integration between a hot and a cold unit.

$$T^{in}\left(s_{inj_h}, p, pp\right) - T^{out}\left(s_{inj_c}, p, pp\right) \geq \Delta T - \Delta T^{U}\left(1 - x\left(s_{inj_h}, s_{inj_c}, p, pp\right)\right), \quad (3\text{-}17)$$
$$\forall\, p, pp \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c}$$

$$T^{out}\left(s_{inj_h}, p, pp\right) - T^{in}\left(s_{inj_c}, p, pp\right) \geq \Delta T - \Delta T^{U}\left(1 - x\left(s_{inj_h}, s_{inj_c}, p, pp\right)\right), \quad (3\text{-}18)$$
$$\forall\, p, pp \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c}$$

Constraints (3-19) to (3-22) ensure that the times at which units are active are synchronized when direct heat integration takes place.

$$tuu\left(s_{inj_h}, p, pp\right) \geq tuu\left(s_{inj_c}, p, pp\right) - M\left(1 - x\left(s_{inj_h}, s_{inj_c}, p, pp\right)\right), \quad (3\text{-}19)$$
$$\forall\, p, pp \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c}$$

$$tuu\left(s_{inj_h}, p, pp\right) \leq tuu\left(s_{inj_c}, p, pp\right) + M\left(1 - x\left(s_{inj_h}, s_{inj_c}, p, pp\right)\right), \quad (3\text{-}20)$$
$$\forall\, p, pp \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c}$$

$$tpp\left(s_{inj_h}, p, pp\right) \geq tpp\left(s_{inj_c}, p, pp\right) - M\left(1 - x\left(s_{inj_h}, s_{inj_c}, p, pp\right)\right), \quad (3\text{-}21)$$
$$\forall\, p, pp \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c}$$

$$tpp(s_{inj_h}, p, pp) \leq tpp(s_{inj_c}, p, pp) + M(1 - x(s_{inj_h}, s_{inj_c}, p, pp)),$$
$$\forall\ p, pp \in P,\ \ s_{inj_h} \in S_{inJ_h},\ \ s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}22)$$

Constraints (3-23) and (3-24) express the starting time of the heating load required for the cold unit and cooling load required for the hot unit at the first interval should be equal to the starting time of the hot and cold unit.

$$tuu(s_{inj_h}, p, pp) = tu(s_{inj_h}, p), \forall\ p, pp \in P,\ \ s_{inj_h} \in S_{inJ_h} \qquad (3\text{-}23)$$

$$tuu(s_{inj_c}, p, pp) = tu(s_{inj_c}, p), \forall\ p, pp \in P,\ \ s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}24)$$

Constraints (3-25) and (3-26) state that the starting time of heating and cooling in an interval should be equal to the finishing time at the previous interval.

$$tuu(s_{inj_h}, p, pp) = tpp(s_{inj_h}, p, pp-1), \forall\ p, pp \in P,\ \ s_{inj_h} \in S_{inJ_h} \qquad (3\text{-}25)$$

$$tuu(s_{inj_c}, p, pp) = tpp(s_{inj_c}, p, pp-1), \forall\ p, pp \in P,\ \ s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}26)$$

Constraint (3-27) ensures that if heat integration occurs, the heat load should have a value that is less than the maximum amount of heat exchangeable. When the binary variable associated with heat integration takes a value of zero, no heat integration occurs and the associated heat load is zero.

$$Ql(s_{inj_h}, s_{inj_c}, p, pp) \leq Q^U x(s_{inj_h}, s_{inj_c}, p, pp), \forall\ p, pp \in P,\ \ s_{inj_h} \in S_{inJ_h},\ \ s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}27)$$

Constraints (3-28) and (3-29) state that if the binary variable associated with heat integration is active, then the binary variable associated with heating and cooling must be active.

$$x(s_{inj_h}, s_{inj_c}, p, pp) \leq y\mathrm{int}(s_{inj_h}, p, pp), \forall\ p, pp \in P,\ \ s_{inj_h} \in S_{inJ_h},\ \ s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}28)$$

$$x(s_{inj_h}, s_{inj_c}, p, pp) \leq y\mathrm{int}(s_{inj_c}, p, pp), \forall\ p, pp \in P,\ \ s_{inj_h} \in S_{inJ_h},\ \ s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}29)$$

Constraints (3-30) and (3-31) state that the heating and cooling loads take on a value for a certain duration when the binary variables associated with heating and cooling are active.

$$tpp(s_{inj_h}, p, pp) - tuu(s_{inj_h}, p, pp) \leq Hy\text{int}(s_{inj_h}, p, pp), \forall\, p, pp \in P,\ s_{inj_h} \in S_{inJ_h} \qquad (3\text{-}30)$$

$$tpp(s_{inj_c}, p, pp) - tuu(s_{inj_c}, p, pp) \leq Hy\text{int}(s_{inj_c}, p, pp), \forall\, p, pp \in P,\ s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}31)$$

Constraints (3-32) and (3-33) state that temperature changes in the heating and cooling units take place when the binary variables associated with heating and cooling are active.

$$T^{in}(s_{inj_h}, p, pp) - T^{out}(s_{inj_h}, p, pp) \leq \Delta T^U(s_{inj_h})y(s_{inj_h}, p, pp),$$
$$\forall\, p, pp \in P,\ s_{inj_h} \in S_{inJ_h} \qquad (3\text{-}32)$$

$$T^{out}(s_{inj_c}, p, pp) - T^{in}(s_{inj_c}, p, pp) \leq \Delta T^U(s_{inj_c})y(s_{inj_c}, p, pp),$$
$$\forall\, p, pp \in P,\ s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}33)$$

Constraint (3-34) states that the cooling of a hot unit will be satisfied by direct heat integration and external cooling utility if required.

$$cl(s_{inj_h}, p) = cw(s_{inj_h}, p) + \sum_{s_{inj_c}} Ql(s_{inj_h}, s_{inj_c}, p, pp),$$
$$\forall\, p, pp \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}34)$$

Constraint (3-35) states that the heating of a cold unit will be satisfied by direct heat integration and external heating utility if required.

$$hl(s_{inj_c}, p) = st(s_{inj_c}, p) + \sum_{s_{inj_h}} Ql(s_{inj_h}, s_{inj_c}, p, pp),$$
$$\forall\, p, pp \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c} \qquad (3\text{-}35)$$

3.3.2 Wastewater minimization model

The superstructure on which the wastewater minimization model is based is depicted in Figure 3.3. Only the water using operations which are part of a complete batch process are depicted. Unit *j* represents a water using operation in which the water used can consist of freshwater, reuse water or reuse and freshwater. Water from unit *j* can be reused elsewhere or sent to effluent treatment.

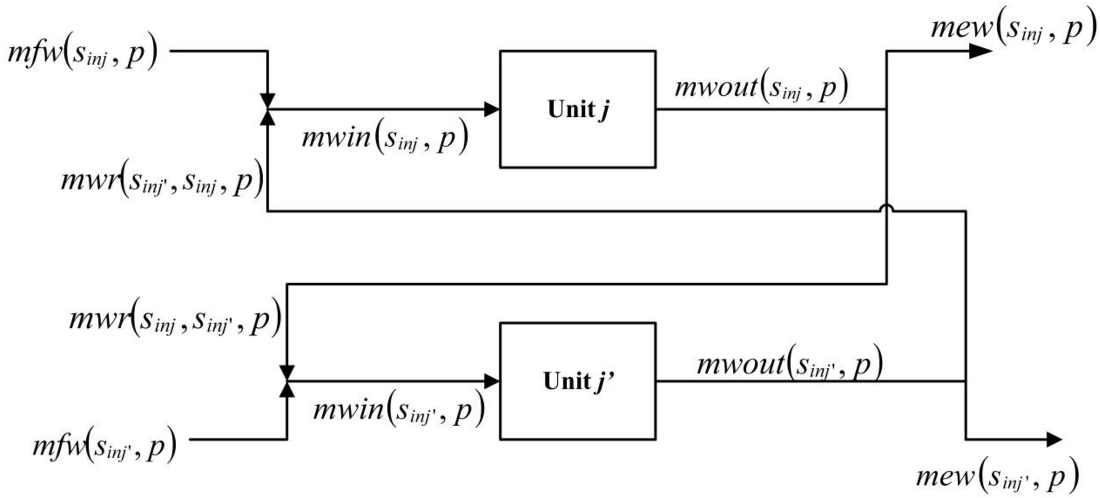


Figure 3.3. Superstructure for water usage.

Constraint (3-36) defines the amount of water entering the unit as the sum of freshwater and reuse water from other units.

$$mwin(s_{inj}, p) = mfw(s_{inj}, p) + \sum_{s_{inj'}} mrw(s_{inj'}, s_{inj}, p), \ \forall\, p \in P,\ s_{inj}, s_{inj'} \in S_{inJ} \qquad (3\text{-}36)$$

Constraint (3-37) states that the amount of water leaving the unit is equal to the sum of reuse water sent to other units and water sent to effluent treatment.

$$mwout(s_{inj}, p) = \sum_{s_{inj'}} mrw(s_{inj}, s_{inj'}, p) + mew(s_{inj}, p), \forall\, p \in P,\ s_{inj}, s_{inj'} \in S_{inJ} \qquad (3\text{-}37)$$

Constraint (3-38) is the water balance around the unit and states that the amount of water entering the unit equals the amount of water leaving the unit.

$$mwin(s_{inj}, p) = mwout(s_{inj}, p), \forall\, p \in P,\ s_{inj} \in S_{inJ} \qquad (3\text{-}38)$$

Constraint (3-39) defines the inlet contaminant load as the mass of contaminant, entering with reuse water.

$$cin(s_{inj}, \Psi, p)\, mwin(s_{inj}, p) = \sum_{s_{inj'}} cout(s_{inj'}, \Psi, p)\, mrw(s_{inj'}, s_{inj}, p), \forall\, p \in P,\ s_{inj}, s_{inj'} \in S_{inJ} \qquad (3\text{-}39)$$

Constraint (3-40) states that the amount of contaminant leaving the unit equals the sum of the contaminant entering into the unit and the contaminant removed from the process.

$$mwout(s_{inj}, p)cout(s_{inj}, \Psi, p) = SMC(s_{inj})mu(s_{inj}, p) + cin(s_{inj}, \Psi, p)mwin(s_{inj}, p), \quad \forall\, p \in P,\; s_{inj} \in S_{inJ} \tag{3-40}$$

Constraint (3-41) ensures that the amount of reused water from unit *j* to other units does not exceed the maximum allowable water in the receiving units. It also indicates whether water from unit *j* is reused or not.

$$mrw(s_{inj}, s_{inj'}, p) \le W_{in}^{U}(s_{inj'})yre(s_{inj}, s_{inj'}, p), \forall\, p \in P,\; s_{inj}, s_{inj'} \in S_{inJ} \tag{3-41}$$

Constraint (3-42) ensures that the reuse of water from unit *j* in other units can occur only if the units are active.

$$yre(s_{inj}, s_{inj'}, p) \le y(s_{inj'}, p), \forall\, p \in P,\; s_{inj}, s_{inj'} \in S_{inJ} \tag{3-42}$$

Constraint (3-43) gives the upper bound on the water entering into unit *j*. It also ensures that water enters into the unit only if it is active.

$$mwin(s_{inj}, p) \le W_{in}^{U}(s_{inj})y(s_{inj}, p), \forall\, p \in P,\; s_{inj} \in S_{inJ} \tag{3-43}$$

In Constraints (3-44) and (3-45), wastewater can only be directly reused if the finishing time of the unit producing wastewater and the starting time of the unit receiving wastewater coincide.

$$tuw(s_{inj'}, p) \ge tpw(s_{inj}, p) - H * yre(s_{inj}, s_{inj'}, p), \forall\, p \in P,\; s_{inj}, s_{inj'} \in S_{inJ} \tag{3-44}$$

$$tuw(s_{inj'}, p) \le tpw(s_{inj}, p) + H * yre(s_{inj}, s_{inj'}, p), \forall\, p \in P,\; s_{inj}, s_{inj'} \in S_{inJ} \tag{3-45}$$

Constraint (3-46) defines the finishing time of the washing operation as the starting time of the washing operation added to the duration of washing.

$$tpw(s_{inj}, p) \ge tuw(s_{inj}, p) + \tau w(s_{inj})y(s_{inj}, p), \forall\, p \in P,\; s_{inj} \in S_{inJ} \tag{3-46}$$

Constraint (3-47) and (3-48) ensure that the starting time of a task in a unit is greater than the finishing time of the washing operations.

$$tu(s_{inj}, p) \geq tpw(s'_{inj}, p-1), \forall\ p \in P,\ s_{inj} \in S^{*}_{inJ} \tag{3-47}$$

$$tu(s_{inj}, p) \geq tpw(s'_{inj}, p-1), \forall\ p \in P,\ s'_{inj} \neq s_{inj},\ s'_{inj}, s_{inj} \in S^{*}_{inJ} \tag{3-48}$$

Constraint (3-49) stipulates that the starting time of the washing operation in a unit occurs after the completion of the task in the unit.

$$tuw(s_{inj}, p) \geq tp(s_{inj}, p), \forall\ p \in P,\ s_{inj} \in S_{inJ} \tag{3-49}$$

Constraints (3-50) and (3-51) ensure that the inlet and outlet concentrations do not exceed the maximum allowable concentration.

$$cin(s_{inj}, \Psi, p) \leq cin^{U}(s_{inj}, \Psi), \forall\ p \in P,\ s_{inj} \in S_{inJ} \tag{3-50}$$

$$cout(s_{inj}, \Psi, p) \leq cout^{U}(s_{inj}, \Psi), \forall\ p \in P,\ s_{inj} \in S_{inJ} \tag{3-51}$$

Constraint (3-52) is the objective function in terms of profit maximization, with profit defined as the difference between revenue from product, cost of utility, raw material cost, freshwater cost and effluent treatment cost.

$$\max \left(\begin{array}{l} \sum_{s^p} price(s^p) q_s(s^p) - \sum_{p} \sum_{s_{injh}} \cos tcw * cw(s_{inj_h}, p) - \\ \sum_{p} \sum_{s_{injc}} \cos tst * st(s_{inj_c}, p) - \sum_{p} \sum_{s_{inj}} \cos tfw * mfw(s_{inj}, p) - \sum_{p} \sum_{s_{inj}} \cos tew * mew(s_{inj}, p) \end{array} \right), \tag{3-52}$$
$$\forall\ p \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c},\ s_{inj} \in S_{inJ}$$

Constraint (3-53) defines minimization of energy and wastewater if the product demand is known.

$$\min \left(\begin{array}{l} \sum_{p} \sum_{s_{injh}} \cos tcw * cw(s_{inj_h}, p) + \sum_{p} \sum_{s_{injc}} \cos tst * st(s_{inj_c}, p) + \\ \sum_{p} \sum_{s_{inj}} \cos tfw * mfw(s_{inj}, p) + \sum_{p} \sum_{s_{inj}} \cos tew * mew(s_{inj}, p) \end{array} \right), \tag{3-53}$$
$$\forall\ p \in P,\ s_{inj_h} \in S_{inJ_h},\ s_{inj_c} \in S_{inJ_c},\ s_{inj} \in S_{inJ}$$

3.4 Case studies

Case studies from published literature were selected to demonstrate the application of the proposed model. The results from the proposed model were obtained using CPLEX 9 as MILP solver and CONOPT 3 as NLP solver in DICOPT interface of GAMS 22.0 and were solved using a 2.4 GHz, 4 GB of RAM, Acer TravelMate 5740G computer.

Case study I

This case study has been investigated extensively in published literature. It is a simple batch plant requiring only one raw material to yield a product as depicted in the state task network (STN) representation in Figure 3.4. The plant comprises of five units and two intermediate storage units. The conversion of the raw material into product is achieved through three sequential processes. The first task can be performed in two units (*j1*and *j2*), the second task can be performed only in unit *j3* and the third task can be performed in units *j4* and *j5*. Task 1 and 2 require cooling during their operation, while task 3 requires heating. The cooling and heating demands are satisfied by external utilities and through heat integration. The operational philosophy requires that the units are cleaned before the next batch is processed. Both freshwater and reuse water can be used as cleaning agents. Table 3.1 gives the capacities of the units, durations of processing and washing tasks, initial availability of states, storage capacities and selling prices and costs for the states. Table 3.2 gives data pertaining to initial and target temperatures for the tasks, specific heat capacities for the states, maximum inlet and outlet contaminant concentrations which are unit dependent and the specific contaminant loads.

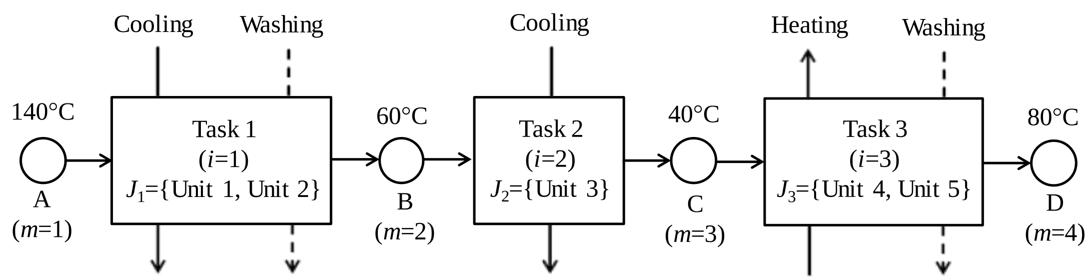


Figure 3.4. STN representation of a simple batch plant producing one product.

Table 3.1. Scheduling data for case study I.

Task (i)	Unit (j)	Max batch size (kg)	Total operation time (h)	Washing time (h)	Material state (m)	Intial inventory (kg)	Max storage (kg)	Revenue or cost ($/kg or $/MJ)
Task 1	Unit 1	100	1.5	0.25	A	1000	1000	0
	Unit 2	150	2	0.3	B	0	200	0
Task 2	Unit 3	200	1.5	0	C	0	250	0
Task 3	Unit 4	100	1	0.25	D	0	1000	5
	Unit 5	150	1.5	0.3	Wash water			0.1
					Waste water			0.05
					Cooling water			0.02
					Steam			1

Table 3.2. Energy and cleaning requirements for case study I.

Task (i)	$T_{s_{inj}}^{in}(^{o}C)$	$T_{s_{inj}}^{out}(^{o}C)$	Unit (j)	C_p(kJ/kg^{o}C)	Max inlet concentration (ppm)	Max outlet concentration (ppm)	Contaminant loading (g contaminant/kg batch)
Task 1	140	60	Unit 1	4	500	1000	0.2
			Unit 2	4	50	100	0.2
Task 2	60	40	Unit 3	3.5	-	-	0.2
Task 3	40	80	Unit 4	3	150	300	0.2
			Unit 5	3	300	2000	0.2
Cooling water	20	30					
Steam	170	160					

Results and discussion

The computational results for case study I using the proposed model for the different scenarios and literature result from Halim and Srinivasan (2011) are presented in Table 3.3. For the scenario without energy and water integration, the total cost of utilities was $293.5. Applying only water integration, the total cost obtained was $259.5, which is an 11.6% reduction, compared to the standalone operation without energy and water integration. For the scenario with energy integration only, a total cost of $208 was obtained, which is a 29.1% reduction compared to the standalone operation. The last column shows the results obtained with combined energy and water integration solved simultaneously giving a total cost of $191.8, which is a 34.6% saving compared to the standalone operation. These results show that in order to achieve the best economic performance, the scheduling problem has to be solved simultaneously considering both water and energy integration. The performance of the proposed model was compared to the sequential optimization technique by Halim and Srinivasan (2011) which resulted in an overall cost of $239.5, which is an 18.4% saving, much less than the 34.6% saving obtained by the proposed model. The efficiency of the proposed model can be attributed to solving the scheduling problem while incorporating water and energy integration in the same framework and using the recent efficient scheduling technique by Seid and Majozi (2012). Figure 3.5 details the possible amount of energy integration between the cold and hot units and the time intervals during which energy integration occurred.

Table 3.3 Computational results for case study I.

	Proposed formulation without water and energy integration	Proposed formulation with water integration	Proposed formulation with energy integration	Halim & Srinivasan (2011) with water and energy integration	Propsed formulation with water and energy integration
Profit ($)	4706.5	4740.5	4791.5	4 764.1	4 808.2
Steam (MJ)	120	120	36.63	43.9	39
Cooling water (MJ)	390	390	281.2	313.9	309
Total freshwater (kg)	1105	878.2	1105	1238.4	977.7
Revenue from product ($)	5000	5000	5000	5000	5000
Cost of steam ($)	120	120	36.63	43.9	39
Cost of cooling water ($)	7.8	7.8	5.623	6.3	6.2
Cost of freshwater ($)	110.5	87.8	110.5	123.8	97.7
Cost of wastewater ($)	55.25	43.9	55.2	61.9	48.9
Total cost ($)	293.5	259.5	208	235.9	191.8
CPU time (s)	2.3	5000	5000	not reported	5000

The energy requirements of unit *j2* and unit *j4* during the interval 3.2 h to 5 h is highlighted to elaborate on the application of the proposed model. The cooling load of unit *j2* between 3.2 h and 4.9 h was 32 MJ. This is partly satisfied through energy integration with unit *j4* in the same time interval, resulting in an external cooling requirement of 26.8 MJ rather than 32 MJ if it operated in standalone mode. At the beginning of the operation of unit *j2* from 3.2 h to 4.25 h, the cooling requirement was 19.76 MJ. This value was obtained using the time average model by multiplying the duration (4.25 h – 3.2 h) and the energy demand per hour (32 MJ/1.7 h (total duration of the task) = 18.823 MJ) where the cooling requirement is fully satisfied by external cooling. For the rest of its operation between 4.25 h to 4.9 h, the cooling requirement was 12.24 MJ, satisfied partly with energy integration (5.2 MJ) and the difference by external cooling. The heating requirement of unit *j4* when it is operated during the interval 4.25 h to 5 h was 6 MJ. From 4.25 h to 4.9 h the steam requirement was 5.2 MJ obtained from the time average model. This heating requirement was fully satisfied during the interval, by integrating with the hot unit *j2*. The rest of the heating, 0.8 MJ, required during its operation between 4.9 h to 5 h was satisfied by external steam.

Figure 3.6 shows the amount of contaminant removed, freshwater usage, amount of reused water and wastewater produced from washing the necessary units. The washing operation of unit *j2* between 4.9 h and 5.2 h required 200 kg of freshwater to remove a contaminant load of 20 g, producing water with a contaminant concentration of 100 ppm. Part of this water produced from unit *j2*, 50 kg, was used for cleaning unit *j4* to remove a contaminant load of 10 g. This was possible because the outlet concentration from unit *j2* (100 ppm) was lower than the maximum inlet contaminant concentration (150 ppm) for unit *j4.* From Figure 3.6 the total amount of reused water was 358.23 kg minimizing the water usage from 1105 kg (without water integration) to 977.7 kg (with water integration). This resulted in a saving of 11.5 % freshwater usage and wastewater produced.

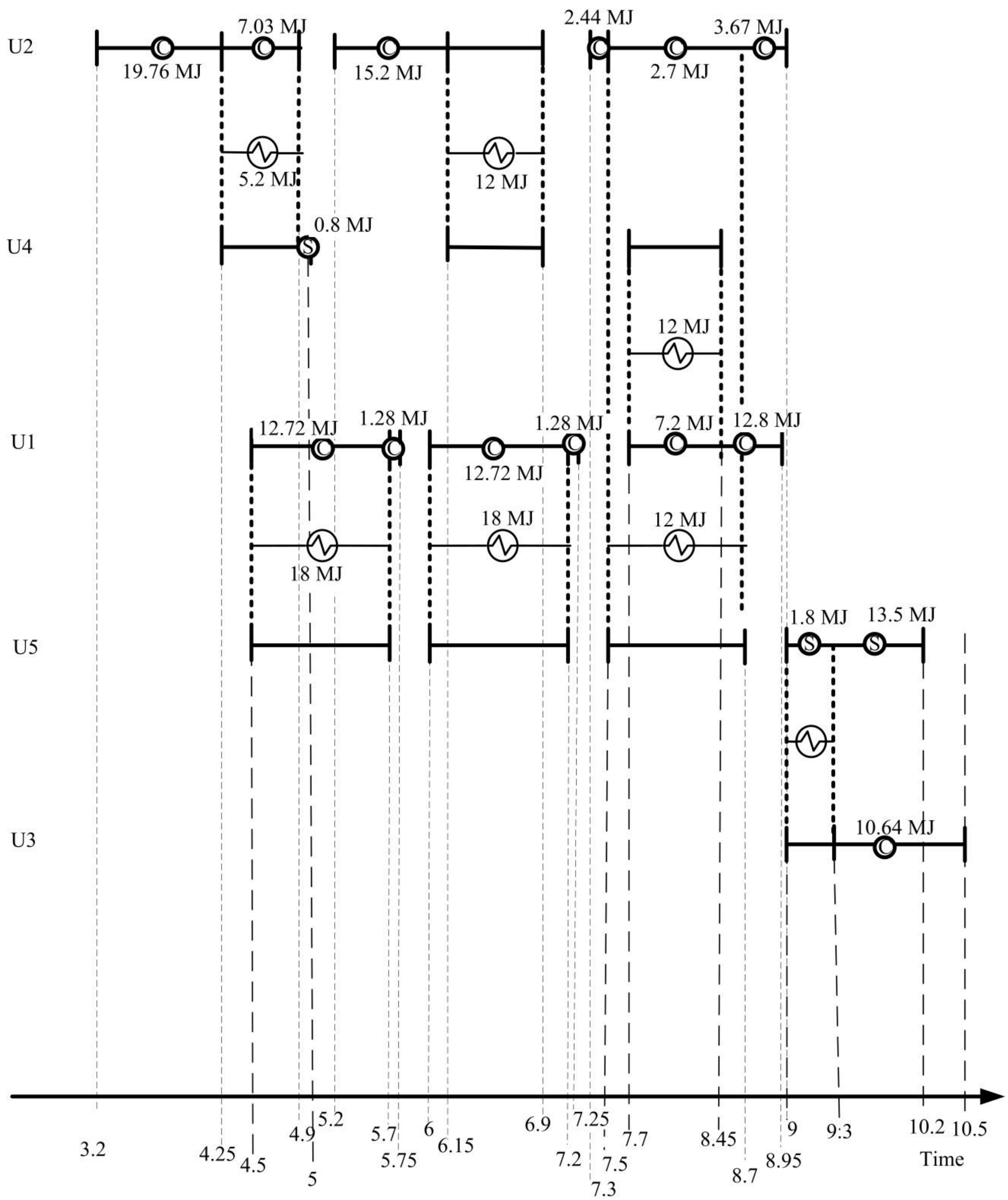


Figure 3.5. energy integration for the time horizon of 12 h for case study I.

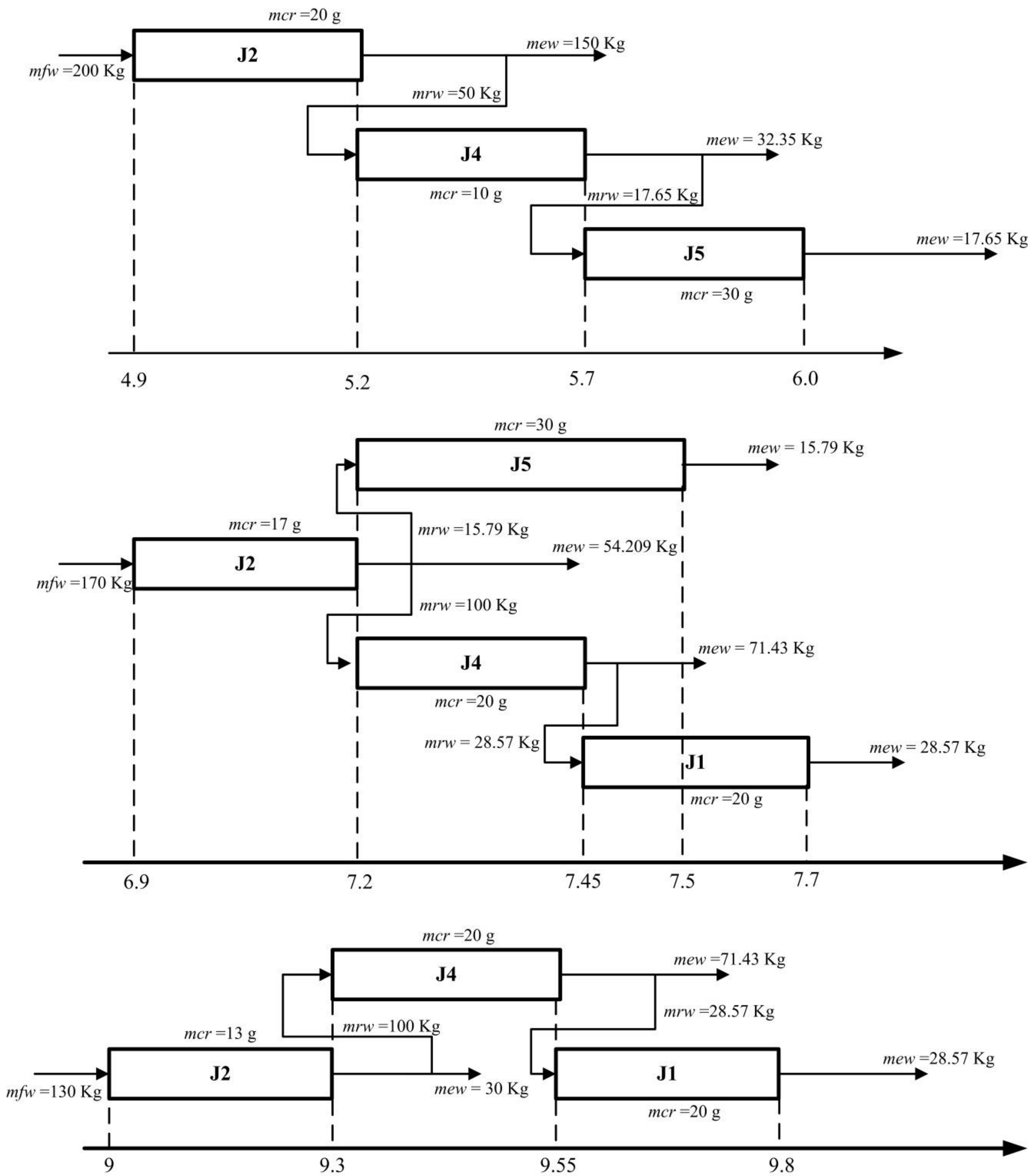


Figure 3.6. Water network with water integration within the time horizon of 12 h for case study I.

The amount of material produced, the starting and finishing times of the processes and washing tasks are shown in Figure 3.7 in the form of a Gantt chart.

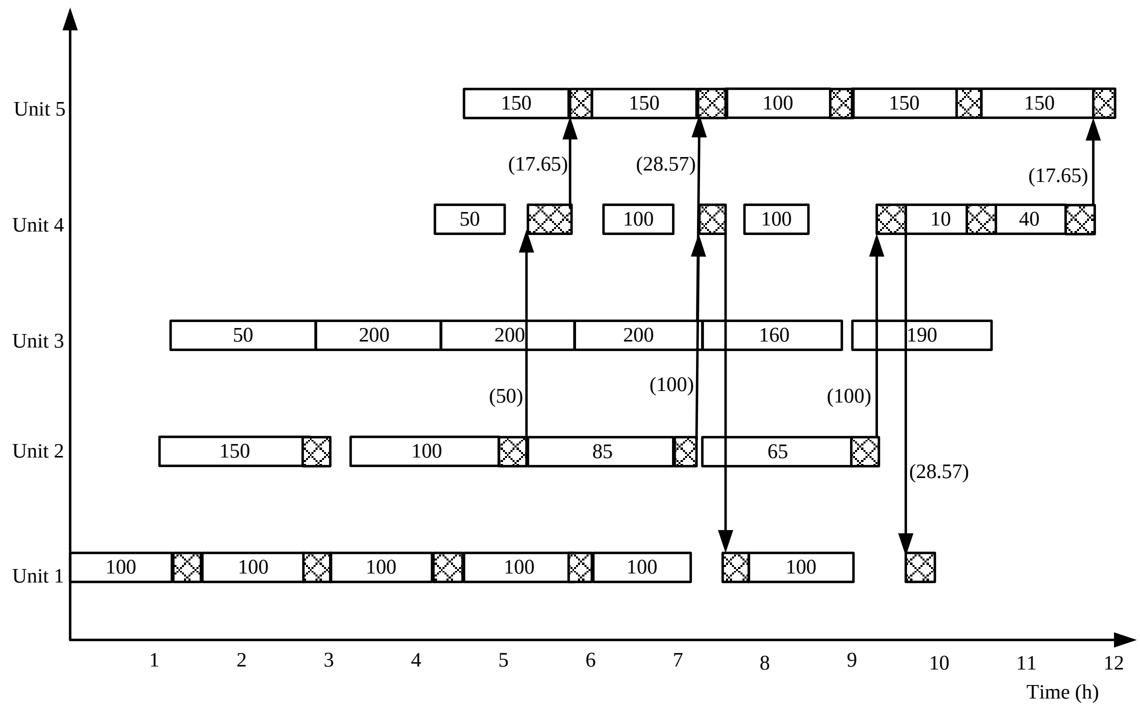


Figure 3.7. Gantt chart for the time horizon of 12 h incorporating energy and water integration for case study I.

Case study II

This case study obtained from Kondili et al. (1993) has become one of the most commonly used examples in literature. However, this case study has been adapted by Halim and Srininvasan (2011) to include energy and water integration. The batch plant produces two different products sharing the same processing units, where Figure 3.8 shows the plant flowsheet. The unit operations consist of preheating, three different reactions and separation. The plant accommodates many common features of multipurpose batch plants such as units performing multiple tasks, multiple units suitable for a task and dedicated units for specific tasks. The STN and SSN representations of the flowsheet are shown in Figure 3.9. Tables 3.4 and 3.5 give the required data to solve the scheduling problem. The production recipe is as follows:

1. Raw material, Feed A, is heated from 50 °C to 70 °C to form HotA used in reaction 2
2. Reactant materials, 50% Feed B and 50% Feed C are used in reaction 1 to produce IntBC. During the reaction the material has to be cooled from 100 °C to 70 °C
3. 60% of the intermediate material, IntBC, and 40% of HotA are used in reaction 2 to produce product 1 and IntAB. The process needs to be heated from 70 °C to 100 °C during its operation
4. 20% of the reactant, Feed C, and 80% of intermediate, IntAB, from reaction 2 are used in reaction 3 to produce ImpureE. The reaction needs its temperature to be raised from 100 °C to 130 °C during its operation
5. The separation process produces 90% product 2 and 10% IntAB from Impure E. Cooling water is used to lower its temperature from 130 °C to 100 °C

The processing time of a task *i* in unit *j* is assumed to be linearly dependent, $\alpha_i + \beta_i B$ on its batch size *B*. Where α_i is a constant term of the processing time of task *i* and β_i is a coefficient of variable processing time of task *i*. The batch dependent processing time makes this case study more complex. Table 3.4 gives the relevant data on coefficients of processing times, the capacity of the processing units, duration of washing, initial inventory of raw materials, storage capacity and relevant costs. Four contaminants are considered in the case study. The maximum inlet and outlet concentrations are given in Table 3.5. The production demand is given as 200 kg for both Prod1 and Prod2. This is a multi-objective optimization problem where the following are optimized: makespan, energy and water consumption.

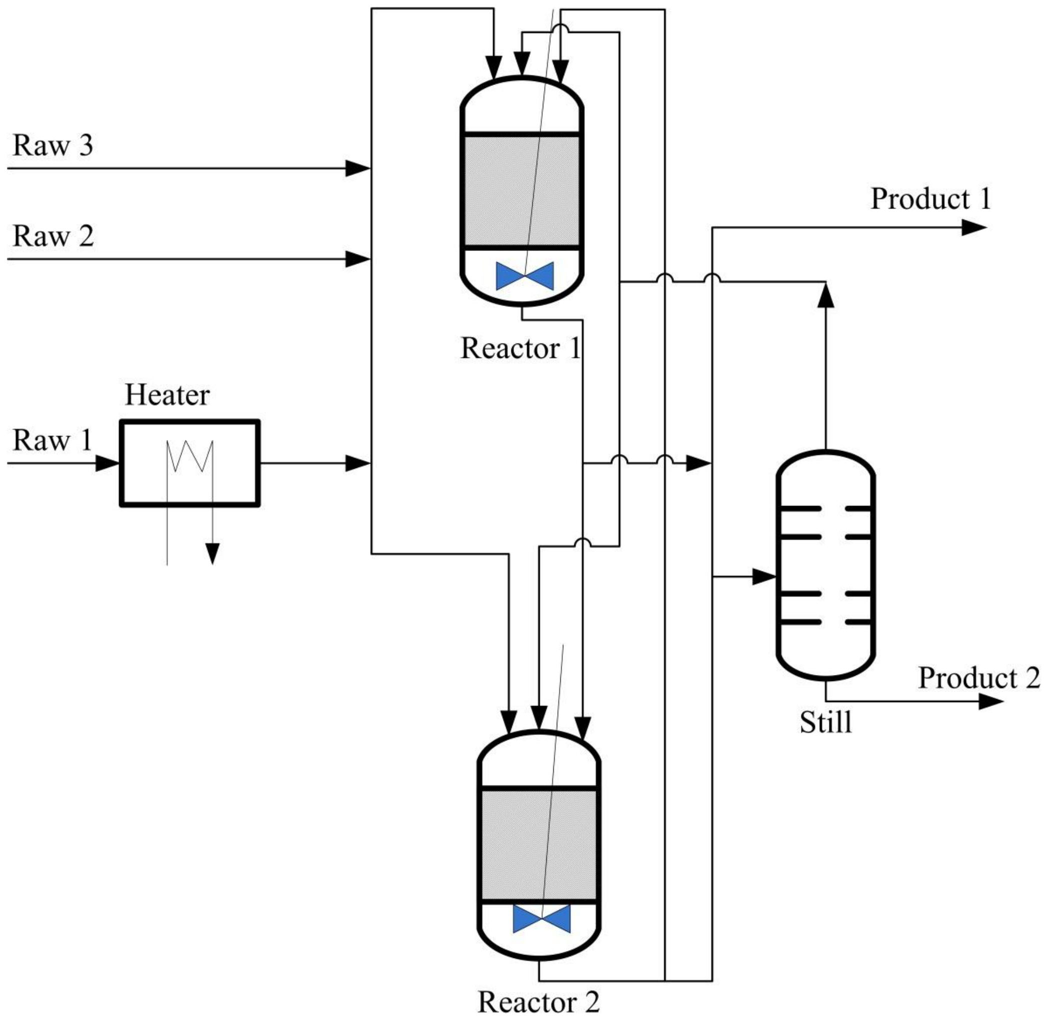


Figure 3.8. Flowsheet for case study II.

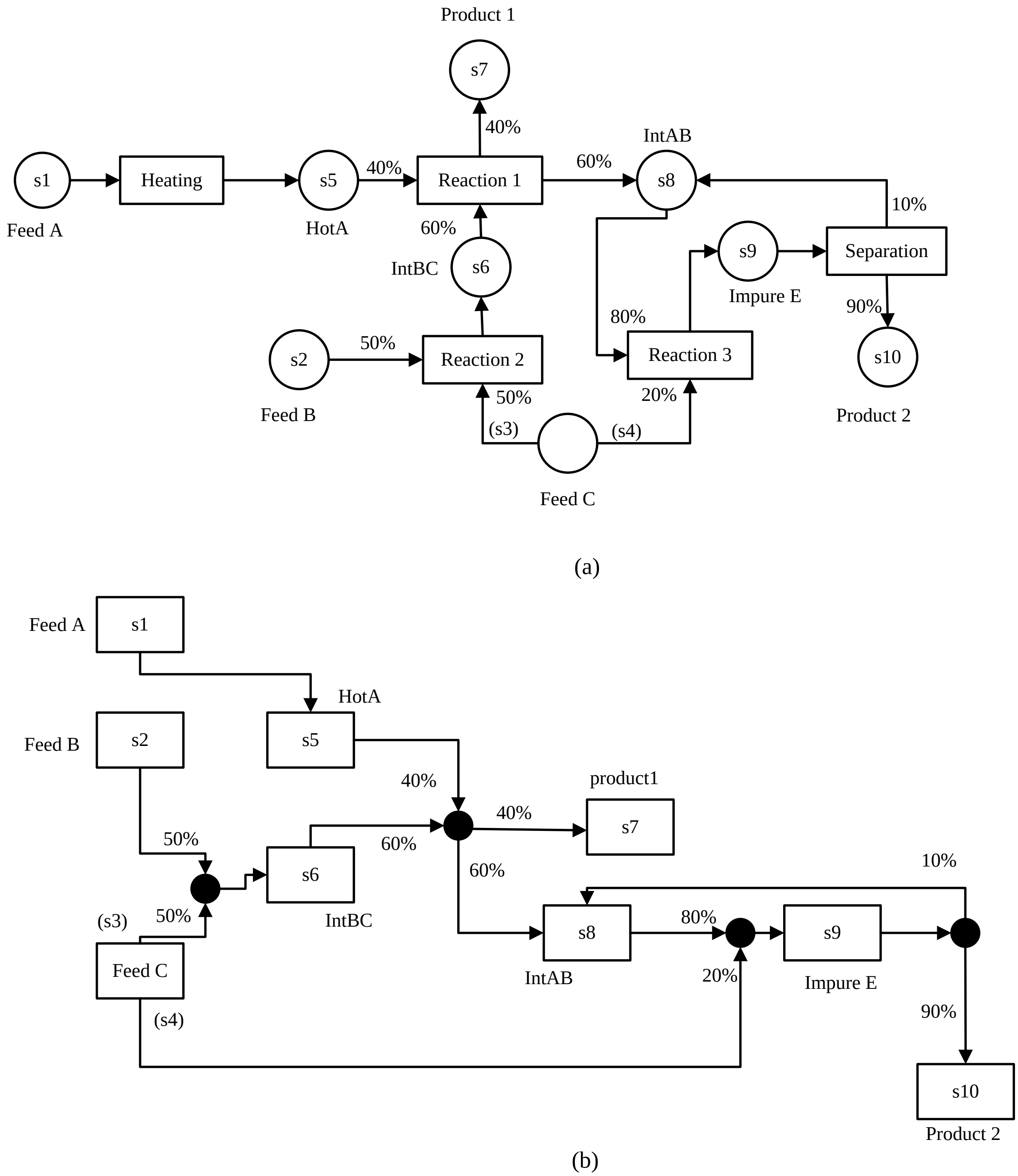


Figure 3.9. STN (a) and (b) SSN representation for case study II.

Table 3.4. Scheduling data for case study II.

Task (i)	Unit (j)	Max batch size (kg)	$\alpha(s_{inj})$	$\beta(s_{inj})$	Washing time	Material state (s)	Initial inventory	Max storage (kg)	Revenue or cost ($/kg or $/MJ)
Heating (H)	HR	100	0.667	0.007	0	Feed A	1000	1000	10
Reaction-1 (R1)	RR1	50	1.334	0.027	0.25	Feed B	1000	1000	10
	RR2	80	1.334	0.017	0.3	Feed C	1000	1000	10
Reaction-2 (R2)	RR1	50	1.334	0.027	0.25	HotA	0	100	0
	RR2	80	1.334	0.017	0.3	IntAB	0	200	0
Reaction-3 (R3)	RR1	50	0.667	0.013	0.25	IntBC	0	150	0
	RR2	80	0.667	0.008	0.3	ImpureE	0	200	0
Separation (S)	SR	200	1.334	0.007	0	Prod1	0	1000	20
						Prod2	0	1000	20
						Wash water			0.1
						Wastewater			0.05
						Cooling water			0.02
						Steam			1

Table 3.5. Data required for energy and water integration.

Task (i)	$T^{in}_{s_{inj}}$ (°C)	$T^{out}_{s_{inj}}$ (°C)	Unit (j)	Cp (kJ/kg°C)	Max inlet concentraion (ppm)				Max outlet concentraion (ppm)				Contaminants (ar,br,cp and dw) loading (g contaminant/ kg batch)
					ar	br	cp	dw	ar	br	cp	dw	
Heating (H)	50	70	HR	2.5									
Reaction-1	100	70	RR1	3.5	300	500	800	400	700	800	1200	900	0.2
			RR2	3.5	300	500	800	400	700	800	1200	900	0.2
Reaction-2	70	100	RR1	3.2	700	600	300	400	1200	1000	600	800	0.2
			RR2	3.2	700	600	300	400	1200	1000	600	800	0.2
Reaction-3	100	130	RR1	2.6	500	200	400	300	800	500	700	900	0.2
			RR2	2.6	500	200	400	300	800	500	700	900	0.2
Separation	130	100	SR	2.8									
Cooling water	20	30											
Steam	170	160											

Results and discussion

The computational statistics for this case study using the proposed model and results obtained by Halim and Srinivasan (2011) are presented in Table 3.6. For makespan minimization an objective value of 19.5 h was obtained using the proposed model, which is better than 19.96 h obtained by Halim and Srinivasan (2011). Using the makespan obtained, the case study was solved using the different scenarios for water minimization, energy minimization and the simultaneous minimization of energy and water by setting customer requirement for Product 1 and Product 2.

The total energy and freshwater required for the standalone operation were 125.5 MJ and 357.94 kg, respectively. For the scenario of water integration only, allowing the use of reuse water the total cost was $112, resulting in 12.2% saving when compared to the standalone operation which had a total cost of $127.52. By using only energy integration the total energy requirement was reduced from 125.5 MJ in standalone operation to 64.56 MJ, resulting in a 48.6% energy saving and a total cost saving of 24.4%.

For the case of simultaneous optimization of energy and water, a significant total cost saving was obtained compared to energy integration alone and water integration alone. A total cost saving of 29.4% was obtained, compared to the standalone operation. The performance of the proposed model was also compared to the technique by Halim and Srinivasan (2011), a total cost of $103 was found using their technique which is higher than $89.96 obtained using the proposed model. Furthermore, the proposed technique is very easy to adopt as opposed to their approach which required 3500 MILP iterations to find the best schedule compared to only 3 MILP iterations in a specified MILP CPU time of 1000 s. This complex case study was solved in a reasonable total specified CPU time of 5000 s.

Table 3.6. Computational results for case study II.

	Proposed formulation without water and energy integration	Proposed formulation with water integration	Proposed formulation with energy integration	Halim & Srinivasan (2011) with water and energy integration	Proposed formulation with water and energy integration
Objective ($)	127.5	112	96.4	103.3	90
Steam (MJ)	75.3	75.3	44.9	61.4	43
Cooling water (MJ)	50.2	50.2	19.7	35.4	17.8
Total freshwater (kg)	357.94	238.1	341.3	275.1	310.9
Revenue from product ($)	4000	4000	4000	4000	4000
Cost of steam ($)	75.3	75.3	44.9	61.4	43
Cost of cooling water ($)	1	1	0.4	0.7	0.4
Cost of freshwater ($)	35.8	23.8	34.1	27.5	31.1
Cost of wastewater ($)	17.9	11.9	17.1	13.8	15.5
Total cost ($)	127.5	112	96.4	103.3	90
Number of time points/slots	11	11	11	N/A	11
CPU time (s)	5,000	5,000	5,000	not reported	5,000

Figure 3.10 shows the Gantt chart related to the optimal usage of energy and water. It also indicates the types of tasks performed in each unit, the starting and finishing times of the processes and washing tasks and the amount of material processed in each batch.

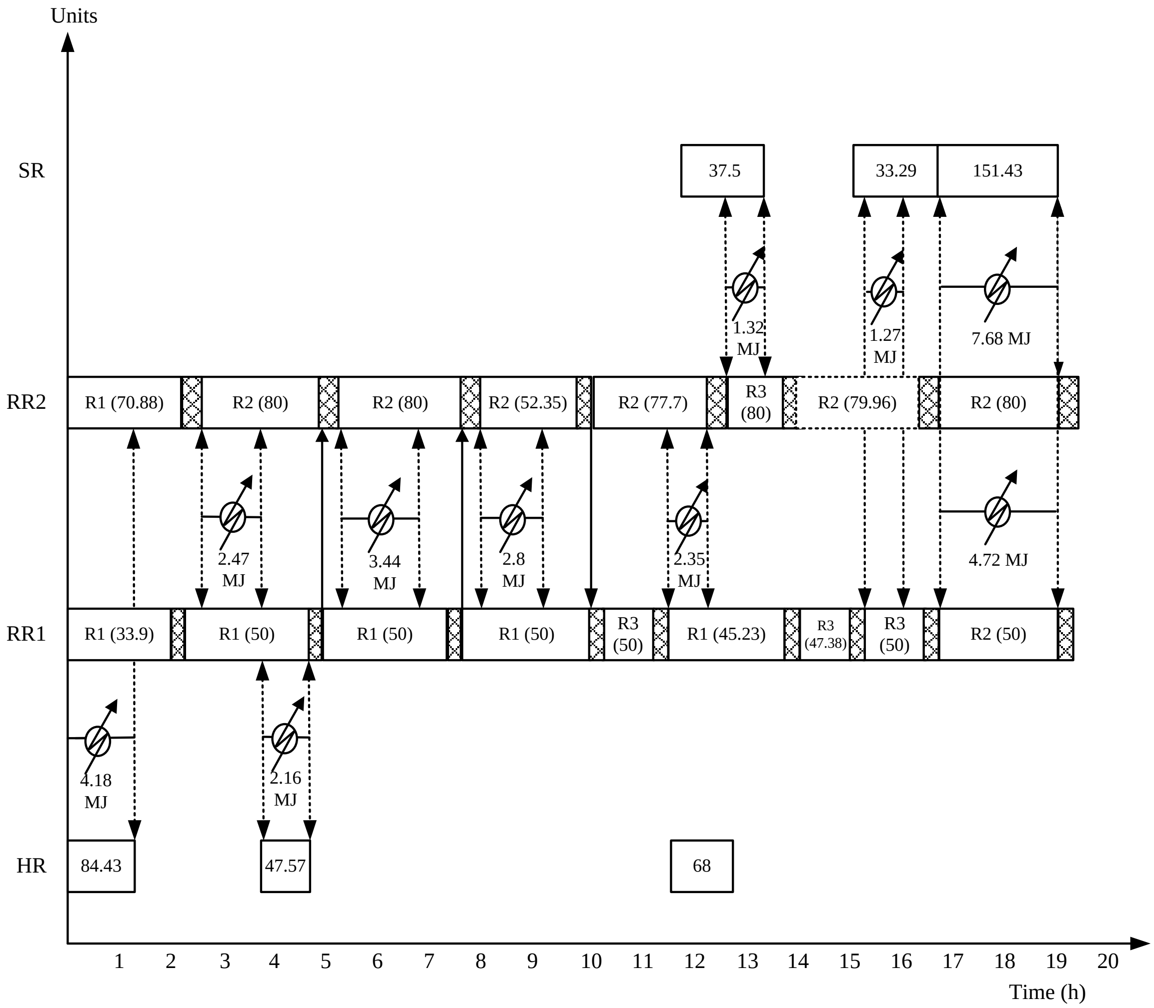


Figure 3.10. Resulting production schedule for case study II with direct heat integration and direct water reuse.

3.5 Conclusions

In the presented method, wastewater minimization and heat integration were both embedded within the scheduling framework and solved simultaneously, thus leading to a truly flexible process schedule. Results from case studies show that profit maximization together with heat integration and wastewater minimization gave a much better overall economic performance. From the case studies a better objective value was achieved using the proposed model compared to a sequential based method.

Nomenclature

Sets

S_{inJ_h}	$\{ s_{inj_h} \mid s_{inj_h}$ task which needs cooling$\}$
S_{inJ_c}	$\{ s_{inj_c} \mid s_{inj_c}$ task which needs heating$\}$
S_{inJ}	$\{ s_{inj} \mid s_{inj}$ any task $\}$
P	$\{ p \mid p$ time point$\}$
S_{inJ_w}	$\{ s_{inj_w} \mid s_{inj_w}$ task which needs washing afterwards$\}$

Parameters

$cp(s_{inj_h})$	Specific heat capacity for the heating task
$cp(s_{inj_c})$	Specific heat capacity for the cooling task
$T^{in}_{s_{injh}}$	Inlet temperature of the heating task
$T^{out}_{s_{injh}}$	Outlet temperature of the heating task
$T^{in}_{s_{injc}}$	Inlet temperature of the cooling task
$T^{out}_{s_{injc}}$	Outlet temperature of the cooling task
ΔT^U	Maximum thermal driving force
ΔT	Minimum thermal driving force
M	Big-M mostly equivalent to the time horizon
Q^U	Maximum heat requirement from the heating and cooling task
$SMC(s_{inj})$	Specific contaminant load produced by a task
$W^U_{in}(s_{inj})$	Maximum water inlet to a processing task
$\tau w(s_{inj})$	Minimum duration required for a washing task

$cin^{Max}(s_{inj}, \Psi)$ Maximum inlet contaminant concentration allowed for contaminant Ψ

$cout^{Max}(s_{inj}, \Psi)$ Maximum inlet contaminant concentration allowed for contaminant Ψ

$price(s^{p})$ Price of a product

H Time horizon of interest

$costfw$ Cost of freshwater

$costew$ Cost of effluent water

$costst$ Cost of steam

$costcw$ Cost of cooling water

Variables

$x(s_{inj_c}, s_{inj_h}, p, pp)$ Binary variable signifying whether heat integration occurs between the hot and cold unit

$y(s_{inj_h}, p)$ Binary variable associated to whether the hot state is active at time point *p* or not

$y(s_{inj_c}, p)$ Binary variable associated to whether the cold state is active at time point *p* or not

$y\text{int}(s_{inj}, p, pp)$ Binary variable associated to whether the hot and cold states are active at time point *p* and extended time point *pp*

$yre(s_{inj}, s_{inj'}, p)$ Binary variable associated with reuse of water from unit *j* to *j'* at time point *p*

$cl(s_{inj_h}, p)$ Cooling load required by the hot task at time point *p*

$hl(s_{inj_c}, p)$ Heating load required by the cold task at time point *p*

$avcl(s_{inj_h}, p)$ Average cooling load required by the hot task at time point *p* using time average model

$avhl(s_{inj_c}, p)$	Average heating load required by the cold task at time point *p* using time average model
$mu(s_{inj_h}, p)$	Amount of material processed by the hot task
$mu(s_{inj_c}, p)$	Amount of material processed by the cold task
$q_s(s^p, p)$	Amount of product stored in a storage unit
$tp(s_{inj}, p)$	Finishing time of a task
$tu(s_{inj}, p)$	Starting time of a task
$clp(s_{inj_h}, p, pp)$	Cooling load required by the hot task active at time point *p* and extended time point *pp*
$hlp(s_{inj_c}, p, pp)$	Heating load required by the cold task active at time point *p* and extended time point *pp*
$tuu(s_{inj}, p, pp)$	Starting time of a task active at time point *p* and extended time point *pp*
$tpp(s_{inj}, p, pp)$	Finishing time of a task active at time point *p* and extended time point *pp*
$T^{in}(s_{inj}, p, pp)$	Inlet temperature of a task active at time point *p* and extended time point *pp*
$T^{out}(s_{inj}, p, pp)$	Outlet temperature of a task active at time point *p* and extended time point *pp*
$Ql(s_{inj_h}, s_{inj_c}, p, pp)$	Amount of heat load exchanged by the hot and cold unit active at time point *p* and extended time point *pp*
$cw(s_{inj_h}, p)$	External cooling water used by the hot task
$st(s_{inj_c}, p)$	External heating used by the cold task
$mwin(s_{inj}, p)$	Mass of water entering to wash a unit after a task is performed
$mwout(s_{inj}, p)$	Mass of water leaving after washing

$mfw(s_{inj}, p)$	Mass of freshwater entering to a unit
$mrw(s_{inj}, s_{inj'}, p)$	Mass of water recycled from unit *j* to another unit *j'*
$mew(s_{inj}, p)$	Mass of water entering to effluent treatment produced from washing
$cin(s_{inj}, \Psi, p)$	Inlet contaminant concentration at time point *p*
$cout(s_{inj}, \Psi, p)$	Outlet contaminant concentration at time point *p*
$tuw(s_{inj}, p)$	Starting time of washing operation for unit *j*
$tpw(s_{inj}, p)$	Finishing time of washing operation for unit *j*

References

Adekola, O., Majozi, T., 2011. Wastewater minimization in multipurpose batch plants with a regeneration unit: Multiple contaminants. *Comput. Chem. Eng*. 35, 2824-2836.

Castro, P.M., Barbosa-Povóa, A.P., Matos, H.A., Novais, A.Q., 2004. Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Ind. Eng. chem. Res*. 43, 105-118.

Cheng, K.F., Chang, C.T., 2007. Integrated water network designs for batch processes. *Ind. and Eng. Chem. Res*. 46, 1241–1253.

Erdirik-Dogan, M., Grossmann, I.E., 2008. Slot-based formulation for the short-term scheduling of multistage, multiproduct batch plants with sequence-dependent changeovers. *Ind. Eng. Chem. Res*. 47, 1159-1163.

Ferrer-Nadal, S., Capón-Garćia, E., Méndez, C.A., Puigjaner, L., 2008. Material transfer operations in batch scheduling. A critical modeling issue. *Ind. Eng. Chem. Res*. 47, 7721-7732.

Floudas, C.A., Lin, X., 2004. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comput. Chem. Eng*. 28, 2109-2129.

Halim, I., Srinivasan, R., 2011. Sequential methodology for integrated optimization of energy and water use during batch process scheduling, *Comput. Chem. Eng*. 35, 1575-1597.

Hui, C.W., Gupta, A., 2000. A novel MILP formulation for short-term scheduling of multi-stage multi-product batch plants. *Comput. Chem. Eng*. 24, 2705-2717.

Ierapetritou, M.G., Floudas, C.A., 1998. Effective continuous-time formulation for short-term scheduling: 1. Multipurpose batch processes. *Ind. Eng. Chem. Res*. 37, 4341-4359.

Janak, S.L., Floudas, C.A., 2008. Improving unit-specific event based continuous time approaches for batch processes: Integrality gap and task splitting. *Comput. Chem. Eng*. 32, 913-955.

Karimi, I.A., McDonald, C.M., 1997. Planning and scheduling of parallel semicontinuous processes. II. Short-term scheduling. *Ind. Eng. Chem. Res*. 36, 2701-2714.

Kondili, E., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations. I. MILP formulation. *Comput. Chem. Eng*. 17, 211-227.

Li, J., Susarla, N., Karimi, I.A., Shaik, M., Floudas, C.A., 2010. An analysis of some unit-specific event-based models for the short-term scheduling of non-continuous processes. *Ind. Eng. Chem. Res*. 49, 633-647.

Lim, M.F., Karimi, I.A., 2003. Resource-constrained scheduling of parallel production lines using asynchronous slots. *Ind. Eng. Chem. Res*. 42, 6832-842.

Liu, Y., Karimi, I.A., 2007. Scheduling multistage, multiproduct batch plants with non identical parallel units and unlimited intermediate storage. *Chem. Eng. Sci*. 62, 1549-1566.

Liu, Y., Karimi, I.A., 2008. Scheduling multistage batch plants with parallel units and no interstage storage. *Comput. Chem. Eng*. 32, 671-693.

Majozi, T., Gouws, J.F., 2009. A mathematical optimisation approach for wastewater minimization in multipurpose batch plants: Multiple contaminants. *Comp. Chem. Eng*. 33, 1826-1840.

Majozi, T., Zhu, X.X., 2001. A novel continuous-time MILP Formulation for multipurpose bach plants. *Ind. Eng. Chem. Res*. 40, 5935-5949.

Maravelias, C.T., Grossmann, I.E., 2003. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Ind. Eng. Chem. Res*. 42, 3056-3074.

Méndez, C.A., Cerdá, J., 2000. Optimal scheduling of a resource-constrained multiproduct batch plant supplying intermediates to nearby end product facilities. *Comput. Chem. Eng*. 24, 369-376.

Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. State of- the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng.* 30, 913-946.

Méndez, C.A., Henning, G.P., Cerdá, J., 2001. An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Comput. Chem. Eng*. 25, 701-711.

Méndez, C.A.; Cerdá, J., 2003. An MILP continuous-time framework for short-term scheduling of multipurpose batch processes under different operation strategies. *Opt. Eng*. 4, 7-22.

Pinto, J.M., Grossmann, I.E., 1994. Optimal cyclic scheduling of multistage continuous multiproduct plants. *Comput. Chem. Eng*. 1994, 18, 797– 816.

Reddy, P.C.P., Karimi, I.A., Srinivasan R., 2004. A new continuous-time formulation for scheduling crude oil operations. *Chem. Eng. Sci*. 59, 1325-1341.

Schilling, G., Pantelides, C., 1996. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Comput. Chem. Eng*. 20, 1221-1226.

Seid, R., Majozi, T., 2012. A robust mathematical formulation for multipurpose batch plants**.** *Chem. Eng. Sci*. 68, 36–53.

Shaik, M., Floudas, C., 2009. Novel unified modeling approach for short term scheduling. *Ind. Eng. Chem. Res*. 48, 2947-2964.

Shaik, M.A., Janak, S.L., Floudas, C.A., 2006. Continuous-time models for short-term scheduling of multipurpose batch plants: A comparative study. *Ind. Eng. Chem. Res.* 45, 6190-4209.

Sundaramoorthy, A., Karimi I.A., 2005. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chem. Eng. Sci*. 60, 2679–702.

Susarla, N., Li, J., Karimi, I.A., 2010. A novel approach to scheduling of multipurpose batch plants using unit slots. *AICHE J.* 56, 1859-1879.

4

CHAPTER 4

Design and Synthesis of Multipurpose Batch Plants Using a Robust Scheduling Platform

Abstract

The increasing interest in multipurpose batch plants is evident, because of their inherent flexibility to cope with an ever changing market environment. These plants are easily reconfigured for product modifications to cover a wide range of operating conditions. They are also suitable to produce different products within the same facility. In spite of this advantage, the design, synthesis and scheduling of multipurpose batch plants can be challenging tasks. This work addresses design, synthesis and scheduling simultaneously. It is known that the scheduling platform employed, has a significant impact on the computational performance of the overall model of the design and synthesis of batch plants. The scheduling model by Seid and Majozi (2012) is extended to incorporate design and synthesis, since it is proven to result in better computational efficiencies. Computational studies are presented to illustrate the effectiveness of the proposed model. A comparison with earlier formulations shows that better computational times and objective functions are obtained.

Keywords: process synthesis, batch plant design, multipurpose batch plant

4.1 Introduction

There is a great deal of interest for manufacturing of fine chemicals, pharmaceutical products, polymers and food and beverage using batch operations because of the advantage of producing low-volume, high quality products, flexibility to adopt complex operations due to fast market change and suitability to manufacture different products using the same facility. In these batch facilities the different products compete for the available resources like equipment, utilities, manpower and storage, which makes the design and operation of this plant a challenging task. The conceptual design problem must determine the number and capacity of the major processing equipment items, utilities and storage tanks so as to meet these design and production objectives at the lowest possible capital and operating cost. The modeling and solution of multipurpose batch processes has received considerable attention in the last two decades. The literature review which follows covers published work under two major headings the basic design and extended design.

4.1.1 Basic grass root design for multipurpose batch plants

Literatures under this category considered simple choice of equipments and associated scheduling. The formulation of Suhami and Mah (1982) was based on the work of Grossmann and Sargent (1979) for multi-product batch plants. A two level problem is derived where the upper level is characterized by the definition of production campaigns using a heuristic procedure, and the lower level includes the design problem with additional constraints resulting in a mixed integer nonlinear programming (MINLP) problem deciding allocation of tasks to units during the time of planning. Klossner and Rippin (1984) looked into the unique assignment case and enumerated all possible product configurations by solving a set partitioning problem followed by the solution of a MINLP for each configuration. The set partitioning problem was further studied by Imai and Nishida (1984) who presented an improved heuristic procedure to solve it. However, no comparative results were presented.

Vaselenak et al. (1987) avoided the two-stage nature of the above approaches by proposing a super- structure embedding all possible product configurations. However, a major difficulty appears to be associated with the derivation of the limiting set of production time constraints, the so called horizon-constraint. A more complete formulation based on a theory of linear inequalities was presented by Faqir and Karimi (1989) where for every case involving a unique production route the equivalent horizon constraints are identified. This work was later extended (Faqir and Karimi, 1990) to multiple production routes assuming unique task-unit allocations with discrete batch size equipment. Cerda et al. (1989) proposed a more general problem than the ones mentioned above. Here, the restrictive assumption that all units within a stage must be allocated to a task if the stage itself is allocated to the same task was relaxed.

Papageorgaki and Reklaitis (1990) proposed a MILP formulation which accommodates equipment used in and out of phase, units available with different sizes in a processing stage and task can be processed in different units. Most of the previous published works were restricted to the following key assumptions: (a) there is a pre specified assignment of equipment items to product tasks, (b) parallel production is only allowed for products that have no common equipment requirements, (c) all units of a given type are identical and can be used in the out-of-phase mode, (d) all units of a given type are devoted to the production of only one product at a time. Shah and Pantelides (1991) presented a formulation incorporating long-term campaign planning. Previous work was limited to all the processing steps involved in the manufacturing of a product taking place within any campaign producing it. This is relaxed with the use of intermediate storage to decouple the manufacturing of each product into several stages each of which can be run independently in campaign mode.

Voudouris and Grossmann (1996) presented a formulation for a special class of sequential multipurpose batch plants where not all the products use the same processing stages. A cyclic schedule was implemented that has the effect of aggregating the number of batches for each product in order to allow the consideration of problems of practical size. It is shown that the no-wait characteristics of substrains can be exploited with a reduction scheme that has the effect of greatly decreasing the dimensionality of the problem. This reduction scheme can be complemented with a tight formulation of the underlying disjunctions in the mixed integer linear programming (MILP) problem to reduce the computational expense.

Xia and Macchietto (1997) presented a general formulation considering linear/nonlinear models for capital cost of processing units/storage vessels and processing task models with fixed, linear and nonlinear processing time in a general form. The formulation resulted in a MINLP solved by a stochastic MINLP optimizer. A short-term scheduling model was implemented based on a discrete time representation which allowed batches of a product to be processed in any sequence.

Barbosa-Povoa and Pantelides (1997) developed a model for single campaign structure with fixed product slate is assumed within a non-periodic operation. A resource-task network (RTN) scheduling framework by Pantelides (1994) was used as a platform. The model allows a detailed consideration of the design problem taking into account the trade-offs between capital costs, revenues and operational flexibility. The optimal solution involves the selection of the required processing and storage equipment items. Lin and Floudas (2001) presented a formulation based on the scheduling formulation of Ierapetritou and Floudas (1998) using a unit specific event point and continuous time representation. The formulation addressed a single campaign production approach where multiple batches of a product can be produced in the given time horizon. A comparison with earlier formulations was given. Heo et al. (2003) developed a three step procedure for the design of multipurpose batch plants using cyclic scheduling. The first MILP model gives the minimum number of equipment units required to produce the products; this configuration is used as an initial plant configuration. The second MILP model determines the minimum cycle time and the third determines the equipment size and scheduling that minimize the cost. The method gave a better solution compared to the separable programming and the evolutionary design method proposed by Fuchino et al. (1994). Castro et al. (2005) presented a general mathematical formulation for the simultaneous design and scheduling of multipurpose plants. The formulation is based on the resource task network process representation and can handle both short-term and periodic problems. The performance of the formulation is illustrated through the solution of two periodic example problems that have been examined in the literature, where the selection and design of the main equipment items and their connecting pipes is considered. They also investigated two example problems by changing demand rates, with the results showing that this set of parameters, which is usually subject to a high degree of uncertainty, affects not only the optimal structure and cost of the plant but also the optimal cycle time. This is a clear indication that both the design and scheduling aspects of the problem should be considered simultaneously. A comparison with an earlier approach of Lin and Floudas (2001) is also presented.

4.1.2 Extended design

Literatures under this category besides the basic design important features such as plant topology and layout have been covered. Barbosa-Povoa and Macchietto (1994) presented for the first time the design of multipurpose batch plants considering plant topology. The formulation is based on the maximal state task network (mSTN) which describes both the recipes and plant possible superstructure. The model optimizes the structural aspect of the plant and the associated production schedule accounting for capital cost of equipment, pipework, operating cost and revenues. Both short term and cyclic scheduling was studied. Penteado and Ciric (1996) and Barbosa-Povoa et al. (2001) addressed layout aspects in the design problems. A 2D small case study was analyzed.

Georgiadis et al. (1997) used a space discretization technique to consider the allocation of equipment items to floor as well as the block layout of each floor. The main drawback of this formulation is a suboptimal solution may result due to the discretization of the available space. Barbosa-Povoa et al. (2002) proposed a general model where both 2D and 3D space was considered. Irregular shapes for each piece of equipment could also be accounted for in the formulation. Patziatsis (2005) formulated a simultaneous layout, design and planning of pipeless batch plants. Barbosa-Povoa (2007) reviewed the design and retrofit of batch plants. The authors included published literature for the last two decades. From the conclusion of the authors there is still a gap in developing efficient techniques addressing large scale problems, detail process operation (cleaning in place, change-over and utility requirements) and plant layout and topology. Pinto et al. (2008) presented a comparative analysis between the STN, m-STN and RTN representations for the design of multipurpose batch plants. A number of problems are solved and compared the formulations based on the different representations.

In this work, a contribution is made by developing an efficient formulation for the design of multipurpose batch plants that consider plant topology. The formulation is posed as a MILP in which the binary variables are the structural choice variables. The proposed model is able to accommodate equipment used in and out of phase, units available in two or more sizes within a processing stage, multiple choices of equipment types for each product task, unit dependent processing time, processing time dependent on batch. The result obtained was compared to the resent published results. The rest of this chapter is organized as follows. A general problem statement for the synthesis and design problem is given in Section 4.2 followed by the mathematical model in Section 4.3. Published case studies are taken and discussed in Section 4.4. Finally, a conclusion is drawn in Section 4.5.

4.2 Problem statement

The optimal plant design can be achieved by developing a model that can solve the following problem:

Given:

- The product recipes (STNs) describing the production of one or more products over a single campaign structure
- The plant flowsheet with all possible equipment units to be installed and the involved connectivity
- The equipment units' suitability to perform the process/storage tasks
- The connections' suitability to transfer materials

- The operating and capital cost data involved in the plant/process installation and operation
- The time horizon of planning
- The production requirements over the time horizon

Determine:

- The optimal plant configuration (i.e. number and type of equipment and the optimal design size of the equipment)
- A process schedule that demonstrates the selected resources to achieve the required production (i.e. the starting and finishing times of all tasks, storage policies, batch sizes, amounts transferred and allocation of tasks to equipment), so as to optimize the economic performance of the plant, measured in terms of the capital expenditure and the operating costs and revenues

4.3 Mathematical formulation

The recent robust scheduling formulation for multipurpose batch plants by Seid and Majozi (2012) is used as a platform for the design and synthesis problem; since it has been proven that the solution performance is dependent on the scheduling framework used.

Equipment existence constraints

Constraint (4-1) defines that in order to execute a task in the unit, the unit must be selected first.

$$\sum_{s_{inj} \in S_{inj}^{*}} y\left(s_{inj}, p\right) \leq e(j), \ \forall \ j \in J, \quad p \in P \qquad (4\text{-}1)$$

Unit size constraints

Constraint (4-2) implies that if the unit is selected the design capacity should be between the minimum and maximum design capacity.

$$V_j^l e(j) \leq ss(j) \leq V_j^u e(j), \ \forall \ j \in J \qquad (4\text{-}2)$$

Capacity constraints

Constraint (4-3) implies that the total amount of all the states consumed at time point *p* is limited by the capacity of the unit which consumes the states and represents lower and upper bounds in capacity of a given unit that processes the effective state.

$$\gamma^{L}_{s_{inj}} ss(j) - \gamma^{L}_{s_{inj}} \left(1 - y\left(s_{inj}, p\right)\right) \leq mu\left(s_{inj}, p\right) \leq \gamma^{U}_{s_{inj}} ss(j),\ \forall\ p \in P,\ j \in J,\ s_{inj} \in S_{inJ} \quad (4\text{-}3)$$

Constraint (4-4) and (4-5) ensure the amount of material stored at any time point *p* is limited by the capacity of the storage.

$$q_s(s, p) \leq s(v),\ \forall\ s \in S,\ p \in P,\ v \in V \quad (4\text{-}4)$$

$$V_v^L eu(v) \leq s(v) \leq V_v^U eu(v),\ \forall\ s \in S,\ p \in P,\ v \in V \quad (4\text{-}5)$$

Material balance for storage

Constraint (4-6) states that the amount of material stored at each time point *p* is the amount stored at the previous time point adjusted by some amount resulting from the difference between state s produced by tasks at the previous time point (*p*-1) and used by tasks at the current time point *p*. This constraint is used for a state other than a product, since the latter is not consumed, but only produced within the process.

$$\begin{aligned} & q_s(s, p) = q_s(s, p-1) - \sum_{j \in J_s^{sc}} muu(s, j, p) + \sum_{j \in J_s^{sp}} muu(s, j, p-1), \\ & \forall\ p \in P,\ p \geq 1,\ \ s \in S \end{aligned} \quad (4\text{-}6)$$

Constraint (4-7) is used for material balance around storage at the first time point.

$$q_s(s, p) = QO(s) - \sum_{j \in J_s^{sc}} muu(s, j, p),\ \forall\ p \in P,\ p = 1,\ \ s \in S \quad (4\text{-}7)$$

Constraint (4-8) states that the amount of product stored at time point *p* is the amount stored at the previous time point and the amount of product produced at time point *p.*

$$q_s\left(s^p, p\right) = q_s\left(s^p, p-1\right) + \sum_{s_{in,j} \in S_{inJ}^p} \rho^{sp}_{s_{in,j}} mu\left(s_{in,j}, p\right),\ \forall\ p \in P,\ \ s^p \in S^p \quad (4\text{-}8)$$

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Material balance around the processing unit

Constraint (4-9) is used to cater for the material processed in the unit and equals the amount of material coming directly coming from the unit producing it and from the storage.

$$\rho_{s_{inj}}^{sc} mu(s_{inj}, p) = muu(s, j, p) + \sum_{j' \in j_s^{sp}} mux(s, j, j', p), \ \forall \ p \in P, \ j \in J, \ s_{inj} \in S_{inJ}, \ s \in S \quad (4\text{-}9)$$

Constraint (4-10) is used to define the amount of material produced at time point *p* which is sent to storage at the same time point *p* and to units that consume it at time point *p*+1.

$$\rho_{s_{in,j}}^{sp} mu(s_{in,j}, p) = muu(s, j, p) + \sum_{j' \in j_s^{sc}} mux(s, j', j, p+1), \quad \forall \ p \in P, \ j \in J, \ s_{inj} \in S_{inJ}, \ s \in S \quad (4\text{-}10)$$

Existence constraints for piping

Constraints (4-11) and (4-12) are applicable to ensure whether pipe connections exist between the processing units and storage as well as to ensure the amount of material transferred through the connection is limited by the design capacity of the connection.

$$V_{j,v}^{L} z(j,v) \le pip(j,v) \le V_{j,v}^{U} z(j,v), \ \forall \ j \in J, \ v \in V \quad (4\text{-}11)$$

$$muu(s, j, p) \le pip(j,v), \ \forall \ j \in J, \ v \in V, p \in P, \ s \in S \quad (4\text{-}12)$$

Constraints (4-13) and (4-14) are similar to (4-11) and (4-12) and are used when the connection is between processing units.

$$V_{j,j'}^{L} w(j, j') \le pipj(j, j') \le V_{j,j'}^{U} w(j, j'), \ \forall \ j \in J \quad (4\text{-}13)$$

$$mux(s, j, j', p) \le pipj(j, j'), \forall \ j, j' \in J, \ p \in P, \ s \in S \quad (4\text{-}14)$$

Duration constraints (batch time as a function of batch size)

Constraint (4-15) describes the duration constraint modeled as a function of batch size where the processing time is a linear function of the batch size. For zero-wait (ZW), only the equality sign is used.

$$tp(s_{inj}, p) \geq tu(s_{inj}, p) + \tau(s_{inj})\, y(s_{inj}, p) + \beta(s_{inj})\, mu(s_{inj}, p),$$
$$\forall\ j \in J,\ p \in P,\ s_{inj} \in S_{inJ} \qquad (4\text{-}15)$$

Sequence constraints

The following two subsections address the proper allocation of tasks in a given unit to ensure the starting time of a new task is later than the finishing time of the previous task.

Same task in same unit

Constraint (4-16) states that a state can only be used in a unit, at any time point, after all the previous tasks are complete. In essence, this implies that a unit must be available before it can be used.

$$tu(s_{inj}, p) \geq tp(s_{inj}, p-1),\ \forall\ j \in J,\ p \in P,\ s_{inj} \in S^{*}_{inj} \qquad (4\text{-}16)$$

Different tasks in same unit

Constraint (4-17) states that a task can only start in the unit after the completion of all the previous tasks that can be performed in the unit. In the context of Constraint (4-17), tasks pertain to different states, hence different tasks in the same unit.

$$tu(s_{inj}, p) \geq tp(s'_{inj}, p-1),\ \forall\ j \in J,\ p \in P,\ s_{inj} \neq s'_{inj},\ s_{inj}, s'_{inj}, \in S^{*}_{inj} \qquad (4\text{-}17)$$

If the state is consumed and produced in the same unit, where the produced state is unstable then in addition to Constraint (4-17), Constraints (4-18) and (4-19) are used.

$$tp(s^{usp}_{inj}, p-1) \geq tu(s^{usc}_{inj}, p) - H(1 - y(s^{usp}_{inj}, p-1)),\ \forall\ j \in J,\ p \in P,\ s^{usc}_{inj} \in S^{usc}_{inj},\ s^{usp}_{inj} \in S^{usp}_{inj} \qquad (4\text{-}18)$$

$$tp(s^{usp}_{inj}, p) \geq tp(s^{usp}_{inj}, p-1),\ \forall\ j \in J,\ p \in P,\ s^{usp}_{inj} \in S^{usp}_{inj} \qquad (4\text{-}19)$$

It should be noted that in this particular situation $tu(s^{usc}_{inj}, p) = tu(s_{inj}, p)$ and $tp(s^{usc}_{inj}, p) = tp(s_{inj}, p)$ Consequently, Constraints (4-17) and (4-18) enforce $tu(s^{usc}_{inj}, p) = tp(s^{usp}_{inj}, p-1)$.

Sequence constraints for different tasks in different units

These constraints state that for different tasks that consume and produce the same state, the starting time of the consuming task at time point *p* must be later than the finishing time of any task at the previous time point *p*-1 provided that the state is used.

If an intermediate state s is produced from one unit.

Constraints (4-20) and (4-21) work together in the following manner:

$$\rho\left(s_{inj}^{sp}\right) mu\left(s_{inj}, p-1\right) \leq q_s(s, p) + V_j^U t(j, p), \forall\ j \in J,\ p \in P,\ s_{inj} \in S_{inJ}^{sp} \quad (4\text{-}20)$$

$$tu\left(s_{inj'}, p\right) \geq tp\left(s_{inj}, p-1\right) - H\left(\left(2 - y\left(s_{inj}, p-1\right) - t(j, p)\right)\right), \forall\ j \in J,\ p \in P,\ s_{inj} \in S_{inJ}^{sp},\ s_{inj'} \in S_{inJ}^{sc} \quad (4\text{-}21)$$

Constraint (4-20) states that if the state *s* is produced from unit *j* at time point *p*-1 but is not consumed at time point *p* by another unit *j'*, i.e. *t*(*j*,*p*) = 0, then the amount produced cannot exceed allowed storage, i.e. $q_s(s,p)$. On the other hand, if state *s* produced from unit *j* at time point *p*-1 is used by another unit *j'* then the amount of state *s* stored at time point *p*, i.e. $q_s(s,p)$ is less than the amount of state *s* produced at time point *p*-1. The outcome is that the binary variable *t*(*j*,*p*) becomes 1 in order for Constraint (4-20) to hold. If the unit performs tasks like separation, distillation and other tasks that produce more than one intermediate at time point *p* then the binary variable *t*(*j*,*p*) becomes *t*(*j*,s,*p*). This allows at the same time point for Constraint (4-21) to be relaxed for the unit that is not using the state produced by unit *j* at time point *p*. Simultaneously, for the other unit that uses the state produced by unit *j* at time point *p* the sequence Constraint (4-21) holds. Constraint (4-21) states that the starting time of a task consuming state *s* at time point *p* must be later than the finishing time of a task that produces state *s* at the previous time point *p*-1, provided that state *s* is used. Otherwise, the sequence constraint is relaxed.

If an intermediate state is produced from more than one unit

Constraint (4-22) states that the amount of state *s* used at time point *p* can either come from storage, or from other units that produce the same state depending on the binary variable *t*(*j*,*p*). If the binary variable *t*(*j*,*p*) is 0, which means that state *s* produced from unit *j* at time point *p*-1 is not used at time point *p*, then Constraint (4-21) is relaxed. If *t*(*j*,*p*) is 1, state *s* produced from unit *j* at time point *p*-1 is used, as a result Constraint (4-21) holds. Although Constraint (4-22) is nonlinear it can be linearized exactly using the Glover transformation developed by Glover (1975).

$$\sum_{s_{inj} \in S_{inJ}^{sc}} \rho_{s_{inj}}^{sc} mu\left(s_{inj}, p\right) \leq q_s\left(s, p-1\right) + \sum_{s_{inj} \in S_{inJ}^{sp}} \rho_{s_{inj}}^{sp} mu\left(s_{inj}, p-1\right) t\left(j, p\right) \quad \forall \; j \in J, \; p \in P \qquad (4\text{-}22)$$

Constraint (4-23) states that a consuming task can start after the completion of the previous task. Constraint(4-23) takes care of proper sequencing time when a unit uses material which is previously stored, that is when the producing task is active at time point *p*-2 and later produces and transfers the material to the storage at time point *p*-1. This available material in the storage at time point *p*-1 is then used by the consuming task in the next time points. This necessitates that the starting time of the consuming task must be later than the finishing time of the producing task at time point *p*-2. Consequently, Constraint (4-23) together with Constraint (4-21) result in a feasible sequencing time when the consuming task uses material, which is previously stored and /or material, which is currently produced by the producing units.

$$tu\left(s_{inj'}, p\right) \geq tp\left(s_{inj}, p-2\right) - H\left(1 - y\left(s_{inj}, p-2\right)\right), \; \forall \; j \in J, \quad p \in P, \quad s_{inj} \in S_{inJ}^{sp}, \; s_{inj'} \in S_{inj'}^{sc} \qquad (4\text{-}23)$$

Sequence constraints for FIS policy

According to Constraint (4-24) and (4-21), the starting time of a task that consumes state *s* at time point *p* must be equal to the finishing time of a task that produces state *s* at time point *p*-1, if both consuming and producing tasks are active at time point *p* and time point *p*-1, respectively.

$$tu\left(s_{inj'}, p\right) \leq tp\left(s_{inj}, p-1\right) + H\left(2 - y\left(s_{inj'}, p\right) - y\left(s_{inj}, p-1\right)\right) \qquad (4\text{-}24)$$
$$\forall \; j \in J, \; p \in P, \; s_{inj} \in S_{inJ}^{sp}, \; s_{inj'} \in S_{inJ}^{sc}$$

Time horizon constraints

The usage and the production of states should be within the time horizon of interest. These conditions are expressed in Constraints (4-25) and (4-26).

$$tu(s_{inj}, p) \leq H\,,\ \forall\ s_{inj} \in S_{inJ}\ , p \in P,\ j \in J \tag{4-25}$$

$$tp(s_{inj}, p) \leq H\,,\ \forall\ s_{inj} \in S_{inJ}\ , p \in P,\ j \in J \tag{4-26}$$

Objective function

Constraint (4-27) is the objective function expressed as maximization of profit. This is obtained from revenue from the sale of products less operating costs for tasks, raw material cost, and capital costs from piping and equipment.

$$\begin{aligned} maximize & \left(\begin{aligned} & \sum_{s^p} price(s^p) q_s(s^p, p) - \sum_p \sum_{s_{inj} \in S_{rm}} CRM * mu(s_{inj}, p) - \\ & \sum_p \sum_{s_{inj} \in S_{inJ}} (FOC * y(s_{inj}, p) + VOC * mu(s_{inj}, p)) \end{aligned} \right) * (AWH/H) \\ & - \sum_{j, \in J} \sum_{v \in V} (CNC * z(j,v) + VCN * pip(j,v)) - \sum_{j' \in J} \sum_{j \in J} (FCNC * w(j,j') + VCNC * pipj(j,j')) \\ & - \sum_{j \in J} (FEC * e(j) + VEQ * ss(j)) - \sum_{v \in V} (FECS * eu(v) + VEQS * s(v)), \forall\ p = P,\ s^p \in S^p \end{aligned} \tag{4-27}$$

Constraint (4-28) is the objective function expressed as minimization of capital and operating cost if the demand for the products is known beforehand within the specified time horizon.

$$\begin{aligned} minimize & \left(\begin{aligned} & \sum_p \sum_{s_{inj} \in S_{rm}} CRM * mu(s_{inj}, p) + \\ & \sum_p \sum_{s_{inj} \in S_{inJ}} (FOC * y(s_{inj}, p) + VOC * mu(s_{inj}, p)) \end{aligned} \right) * (AWH/H) \\ & + \sum_{j, \in J} \sum_{v \in V} (CNC * z(j,v) + VCN * pip(j,v)) + \sum_{j' \in J} \sum_{j \in J} (FCNC * w(j,j') + VCNC * pipj(j,j')) \\ & + \sum_{j \in J} (FEC * e(j) + VEQ * ss(j)) + \sum_{v \in V} (FECS * eu(v) + VEQS * s(v)), \forall\ p = P,\ s^p \in S^p \end{aligned} \tag{4-28}$$

4.4 Case studies

In order to demonstrate the applicability of the proposed model, two published literature examples are presented and discussed. The results in all the case studies for the proposed model were obtained using CPLEX 9.1.2/ GAMS 22.0 on a 2.4 GHz, 4 GB of RAM, Acer TravelMate 5740G computer. The computational results for the literature models are taken directly from the literature for comparison.

Case I

At first this case study was studied by Kondili et al. (1993) and it has become one of the most common examples that appearing in literature. This is a typical multipurpose batch plant that produces two different products sharing the same processing units. A task can be conducted in multiple units, a unit can perform different tasks and the products can take different production lines. The unit operations consist of preheating, three different reactions and separation. The STN representation of the flowsheet is shown in Figure 4.1. Full connectivity between equipment is assumed. The plant superstructure is shown in Figure 4.2. The data pertaining to equipment and materials are given in Table 4.1 and 4.2, respectively.

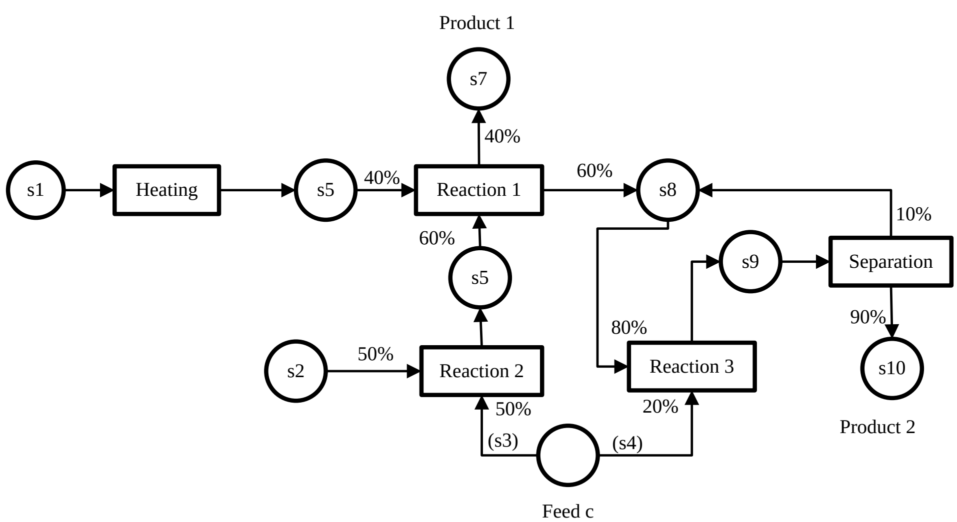


Figure 4.1. Recipe representation for Case I.

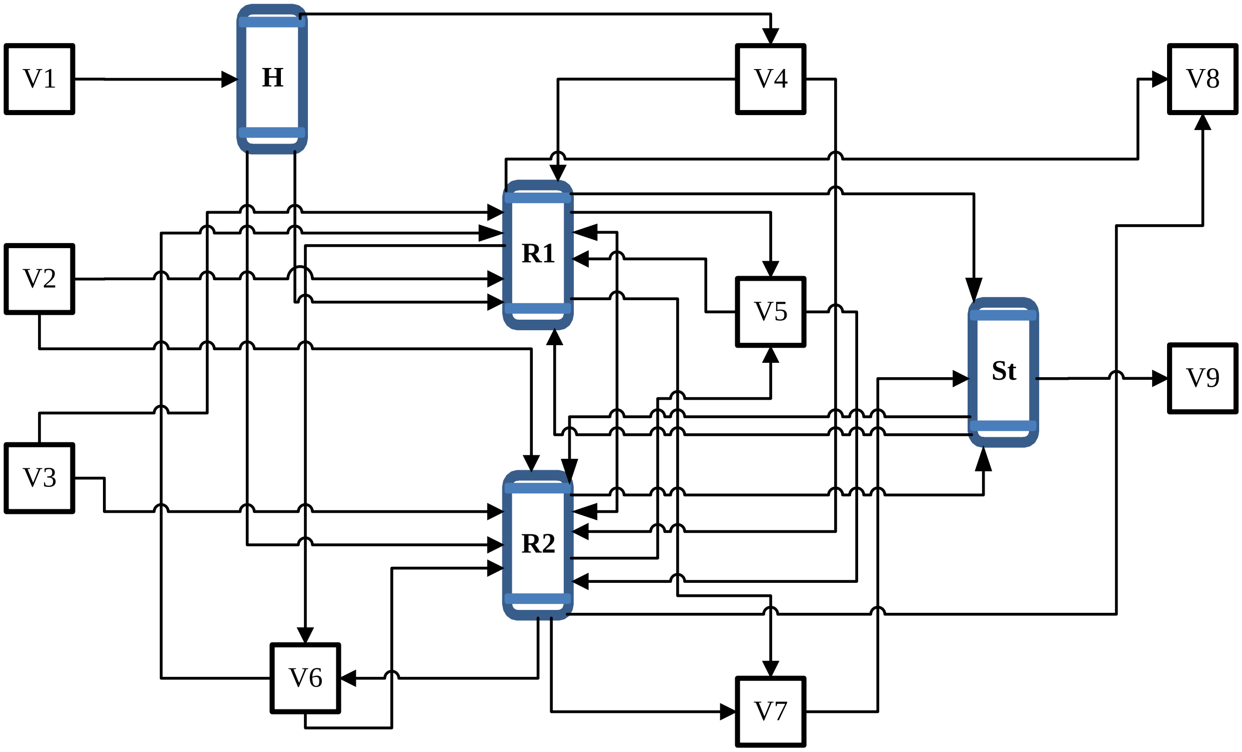


Figure 4.2. Superstructure for Case I.

Table 4.1. Equipment data for Case I.

unit	Capacity	Suitability	processing time	Cost model (k$)
Heater	20.0-50.0	Heating	1.0+0.0067*b*	100.0+0.2*s*
Reactor 1	50.0-70.0	Reaction 1	2.0+0.0267*b*	150.0+0.5*s*
		Reaction 2	2.0+0.0267*b*	
		Reaction 3	1.0+0.0133*b*	
Reactor 2	70	Reaction 1	2.0+0.0167*b*	120
		Reaction 2	2.0+0.0167*b*	
		Reaction 3	1.0+0.0083*b*	
Still	50.0-70.0	Separation	2.0+0.0033*b*	150.0+0.3*s*
Vessel 4	10.0-30.0	(Hot A)		30.0+0.1*s*
Vessel 5	10.0-60.0	(Int BC)		15.0+0.1*s*
Vessel 6	10.0-70.0	(Int AB)		10.0+0.1*s*
Vessel 7	50.0-100.0	(Impure E)		20.0+0.2*s*

Table 4.2. Material data for Case I

State	Storage capacity	Price (k$)	Requirment
Feed A	Unlimited	-0.001	0
Feed B	Unlimited	-0.002	0
Feed C	Unlimited	-0.0015	0
Hot A	Vessel 4	0	0
Int BC	Vessel 5	0	0
Int AB	Vessel 6	0	0
Impure E	Vessel 7	0	0
Product 1	Unlimited	0.02	40
Product 2	Unlimited	0.03	60

Results and discussion

The results obtained for Case I are presented in Table 4.3. The proposed model requires 4 event points, 52 binary variables, 227 continuous variables and 572 constraints and is solved in less than one second. Compared to other literature models the proposed model gave less model size and computational time to solve. An objective value of 569.18 (k$) was obtained in this work, which is better than the objective values obtained by other formulations.

Table 4.3. Computational statistics for Case I using the different formulations.

Model	Time points	Integer variables	Continuous variables	Constraints	Obj MILP (K$)	CPU_s
Xia and Macchietto (1997)	8	288^o	201^o	425^o	-	-
	8	62^t	34^t	122^t	585.6	2407
Lin and Floudas (2001)	6	128	341	877	572.9	22.5
Castro et.al (2005)	7	74	294	395	572.7	6.6
This work	4	52	227	572	569.2	0.08

t = transformed formulation, o = original formulation

The proposed model selected all the processing units; however, it avoided the necessity of all the intermediate storage units given in the superstructure and as a result led to less capital investment compared to other formulations, which required an intermediate storage for IntAB. Figure 4-3 shows the optimal Gantt chart in achieving the production requirement for the time horizon of 12 h with the corresponding equipment size. The optimal plant structure obtained by the proposed model and by Lin and Floudas (2001) is depicted in Figure 4.4.

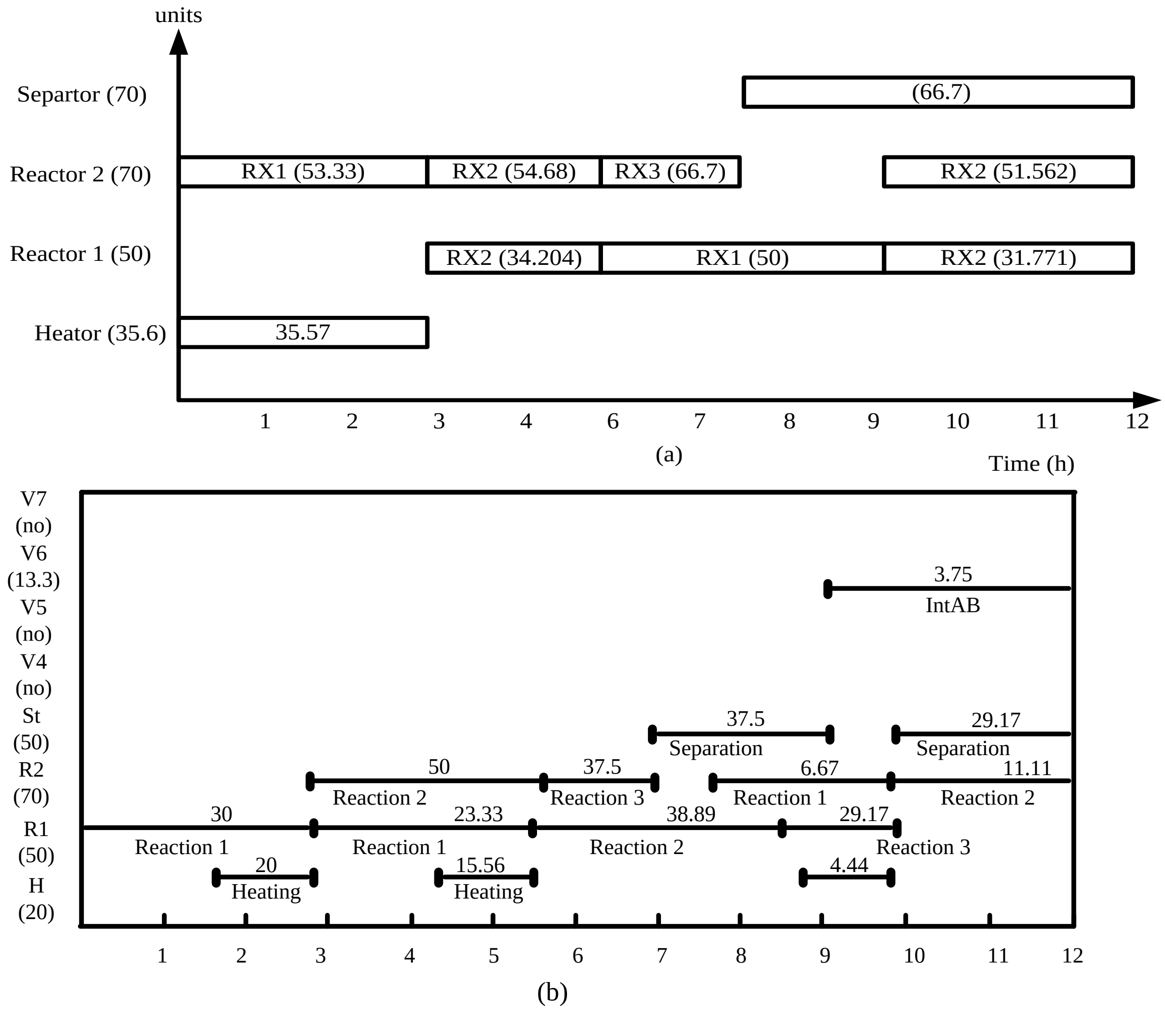
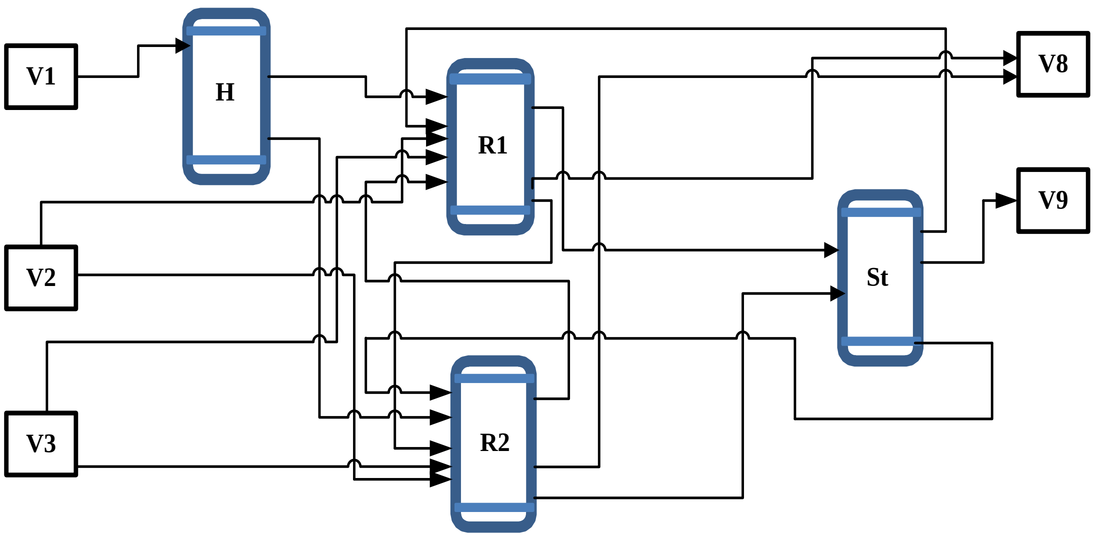
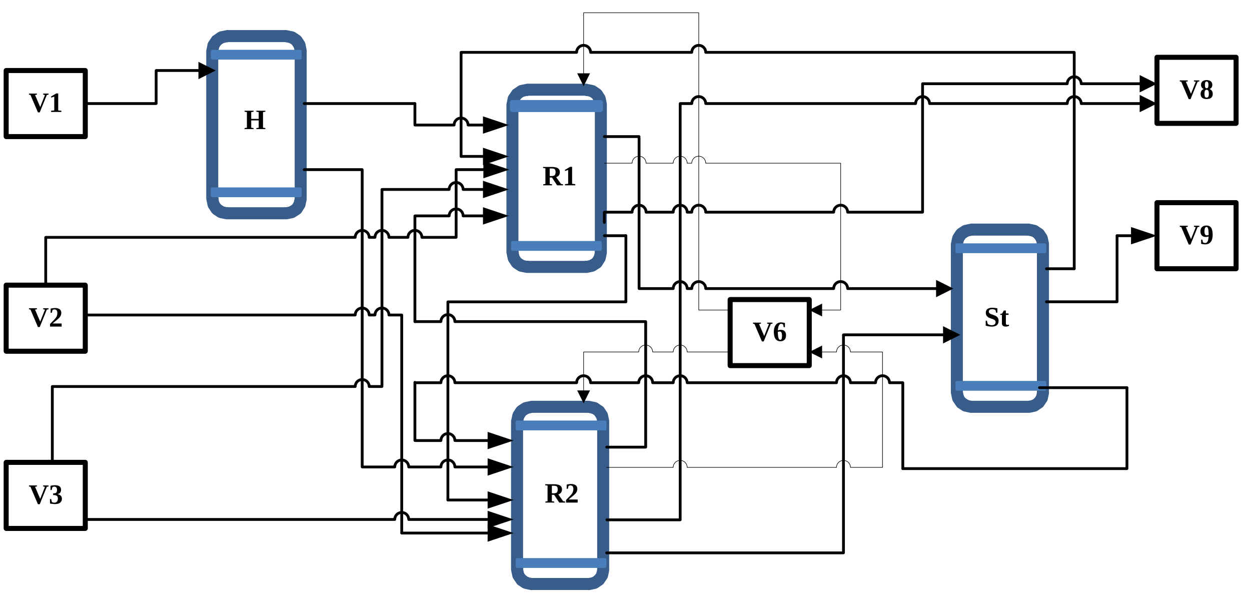


Figure 4.3. (a) Gantt chart using the proposed model, (b) Gantt chart using the formulation of Lin and Floudas (2001).



(a)



(b)

Figure 4.4. (a) Optimal structure using the proposed model, (b) optimal structure using the formulation of Lin and Floudas (2001).

Case II

This case study was taken from the petrochemical industry (Kallrath, 2002) and was later used as a benchmark problem in the scheduling environment for multipurpose batch plants. Pinto et al. (2008) adopted the case study for the design and synthesis problem. The maximization of profit that results in optimal design and synthesis of the plant is considered for the production time horizon of 120 h. The recipe representation for the plant is presented in Figure 4.5. It is a complex problem where 10 reactors, 15 storage units and 18 states are considered in the superstructure presented in Figure 4.6. There are reactors dedicated to a specific task while there are also reactors conducting multiple tasks some capable of executing up to 5 different tasks. Mixed intermediate storage (MIS) policy is assumed. Piping cost from equipment connections is considered, with the connections capacity ranging from 0 to 300 m.u. and have associated fixed/variable cost of 0.1/ 0.01(10^3)c.u. Operating costs for the tasks are fixed/variable of 0.1/ 0.01(10^3)c.u. A raw material costs of 0.002 (10^3)c.u. per m.u. is given. Equipment data is obtained in Table 4.4. The plant is required to satisfy demand requirement (30 tons for S14, S15 and S16, 20 tons for S17 and 40 tons for S18) in a time horizon of 120 h. The selling price for each product is 1000 c.u. per m.u. It is assumed that the plant operates for 366 days.

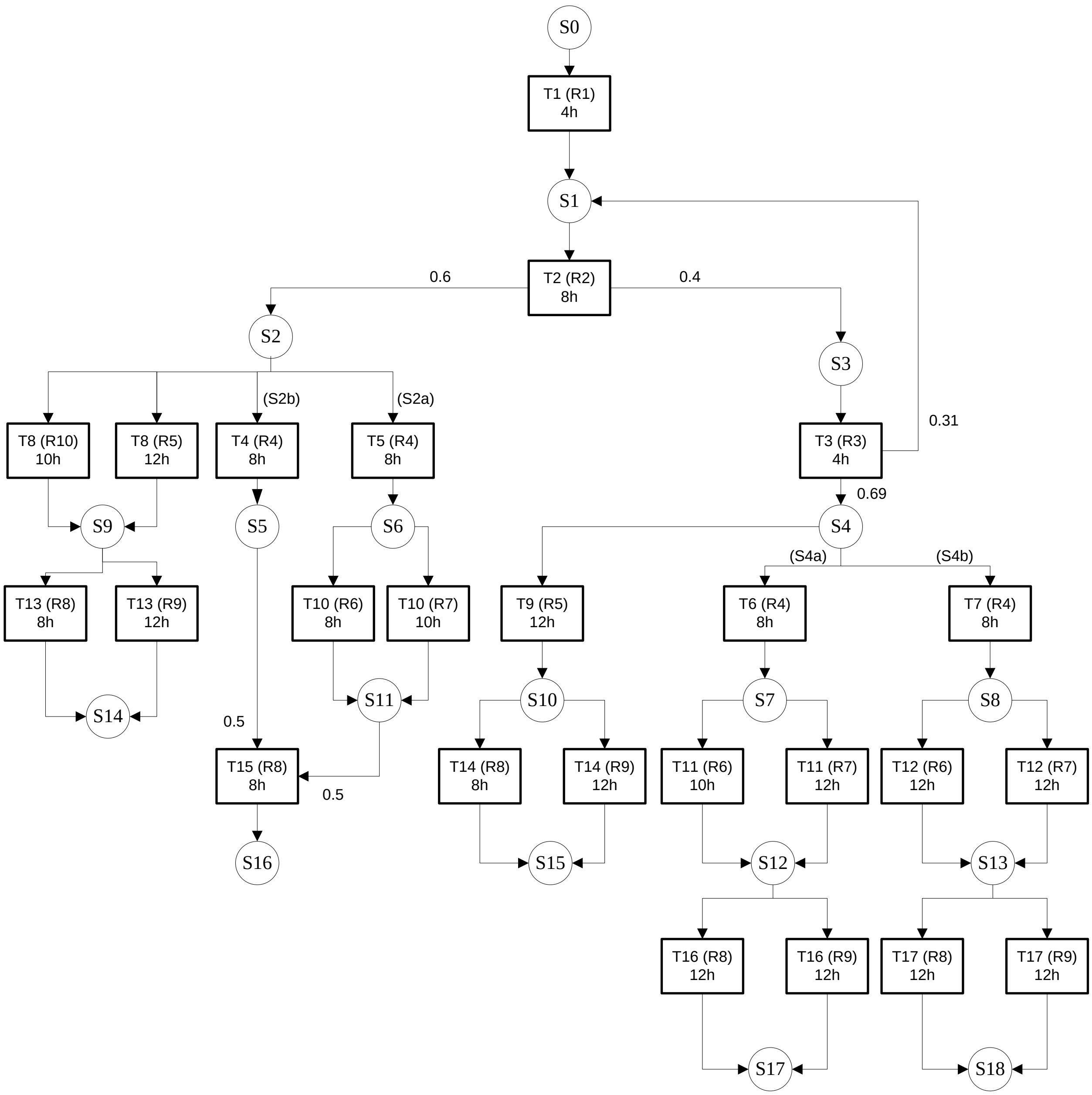


Figure 4.5. STN representation for Case II.

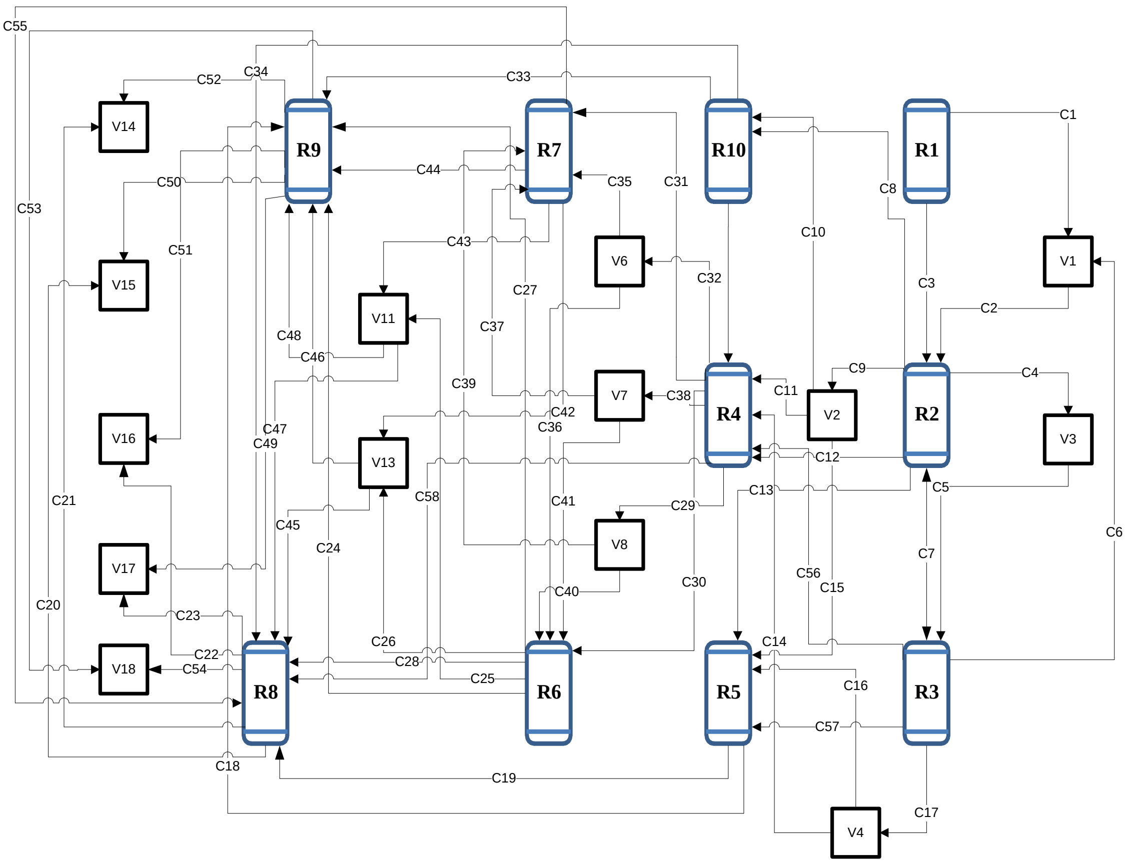


Figure 4.6 Superstructure of the plant for Case II.

Table 4.4. Equipment data for Case II.

Unit	Suitability	Capacity (m.u)	Costs (10^3 c.u. fix:var)
R1	Task T1	0:100	30:0.5
R2	Task T2	0:100	30:0.5
R3	Task T3	0:500	30:0.5
R4	Task T4, T5, T6, T7	0:300	30:0.5
R5	Task T8, T9	0:300	30:0.5
R6	Task T10, T11, T12	0:300	30:0.5
R7	Task T10, T11, T12	0:300	30:0.5
R8	Task T13, T14, T15, T16, T17	0:300	30:0.5
R9	Task T13, T14, T15, T16, T17	0:300	30:0.5
R10	Task T8	0:300	30:0.5
V0	Store S0	0:1000	0.1:0.01
V1	Store S1	0:1000	0.1:0.01
V2	Store S2	0:1000	0.1:0.01
V3	Store S3	0:1000	0.1:0.01
V4	Store S4	0:1000	0.1:0.01
V6	Store S6	0:500	0.1:0.1
V7	Store S7	0:500	0.1:0.1
V8	Store S8	0:500	0.1:0.1
V11	Store S11	0:500	0.1:0.1
V13	Store S13	0:500	0.1:0.1
V14	Store S14	0:500	0.1:0.1
V15	Store S15	0:500	0.1:0.1
V16	Store S16	0:500	0.1:0.1
V17	Store S17	0:500	0.1:0.1
V18	Store S18	0:500	0.1:0.1

Results and discussion

Table 4.5 gives the computational statistics for Case II using the different formulations. The results for the literature models are taken directly from the paper of Pinto et al. (2008). The results obtained clearly differentiate the different formulation efficiencies. The adopted STN and RTN formulations of Pinto et al. (2008) led to a significant reduction in the model size from their classical STN and RTN representations (26147 variables for STN-adopted vs. 31547 variables for STN, 33405 variables for RTN-adopted, 115305 variables for RTN). The m-STN representation gives fewer variables and requires less computational time compared to STN-adopted and RTN-adopted.

Table 4.5. computational statistics for Case II.

Methodology	Objective value	Total variables	Binary variables	Constraints	CPU times (s)
Pinto et al. (2008)					
STN	2.1 (10^7)	31547	10828	59728	11975.6
STN-adapted	2.1 (10^7)	26147	9028	51328	1445.4
m-STN	2.1 (10^7)	17508	3147	29822	1056.9
RTN	2.1 (10^7)	115305	55677	155653	12812.9
RTN-adapted	2.1 (10^7)	33405	9074	54136	1009.1
This work	6.92 (10^7)	4485	776	8895	111

The proposed formulation significantly reduces the number of binary and continuous variables required compared to other formulations, requiring 776 binary variables and 4485 total variables. The current formulation required 16 time points and solved in a CPU time of 111 s to a 5% of margin of optimality for fair comparison since the literature models were also solved with the same margin of optimality. It is worth mentioning that a new objective value of 6.92 (10^7) was obtained; which is a great improvement (228.6%) compared to the objective value, 2.1 (10^7) obtained by other formulations. Within the time horizon of 120 h the proposed formulation produced more products compared to other formulations (480 m.u. for S14, 243 m.u. for S15, 240 m.u. for S16, 20 m.u. for S17 and 40 m.u. for S18 obtained by this work vs. 60 m.u. for S14, 60 m.u. for S15, 50 m.u. for S16, 39 m.u. for S17 and 78 m.u. for S18 obtained by other formulations). The efficiency of the model is attributed to the use of the recent robust scheduling platform of Seid and Majozi (2012) for the design and synthesis problem. The optimal equipment selection and size obtained by this work is obtained in Table 4.6. The formulation avoids having reactors 7 and 10 and storage units V7, V8, and V13 available in the superstructure.

Table 4.6. Optimal equipment design capacity for Case II.

Equipment	Capacity (m.u.)
V0	1088.4
R1, R2, V1, C2, C3	100
R3, V3, C5	280
R4, V11, C11, C47, C58	120
R5, R8, V2, V16, C15, C21, C22, C19	240
R6, V18, C4, C14, C25, C53, C24, C30	40
R9, C50, C18	110.4
V4	173.2
V6, C32, C36	26.7
V14	480
V15	243.6
V17, C49	20
C6	86.8
C9	60
C17	193.2
C16, C20	133.2

The optimal superstructure that depicts the selected equipments and the associated connectivity is given in Figure 4.7. The Gantt chart that detailed the amount of material processed, the type of task performed and the starting and finishing times for each piece of equipment using the proposed formulation is given in Figure 4.8. For comparison the Gantt chart obtained using the formulation of Pinto et al. (2008) is also presented in Figure 4.9. This work also explicitly addressed the allocation of materials in the plant. Figure 4.10 depicts the amount of material and its location in the plant at any given time point p for state $S6$. The amount of material state S6 produced from unit R4 at time point $p4$, $p6$ and $p9$ was sent to storage at time 28 h, 44 h and 60 h respectively. The storage sent state S6 to the consuming unit R6 at time 28 h, 56 h and 64 h. The storage holds the state S6 for the time interval from 44 h – 56 h and 60 h – 64 h only, during the entire time horizon. Direct transfer of material state S6 also occurred from the producing unit to consuming unit; where the amount of material processed by unit R4 at time point $p6$ produced state S6 at the time 44 h and transfers to the consuming unit R6 at the next time point $p7$ where it starts processing state S6 at the same time of 44 h. Consequently, the model allows one to know the exact location of materials and their amount in the plant at any given time.

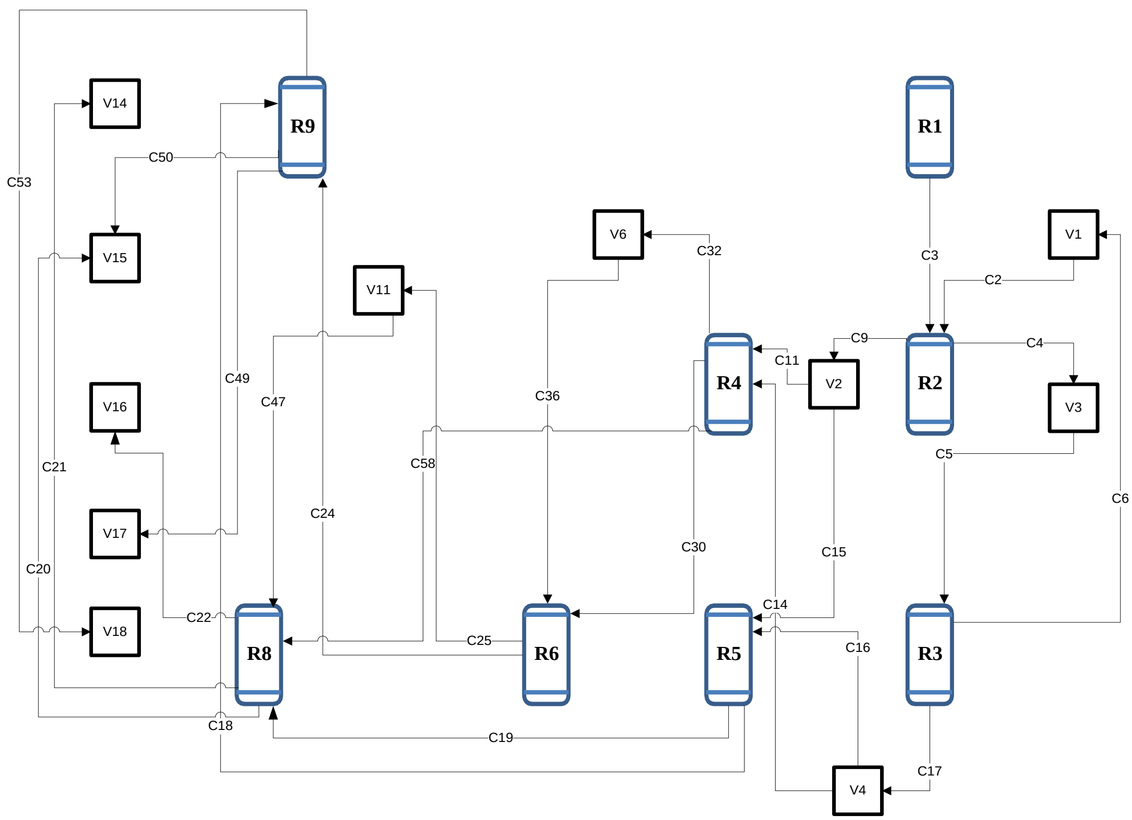


Figure 4.7. Optimal plant structure and associated pipe connectivity.

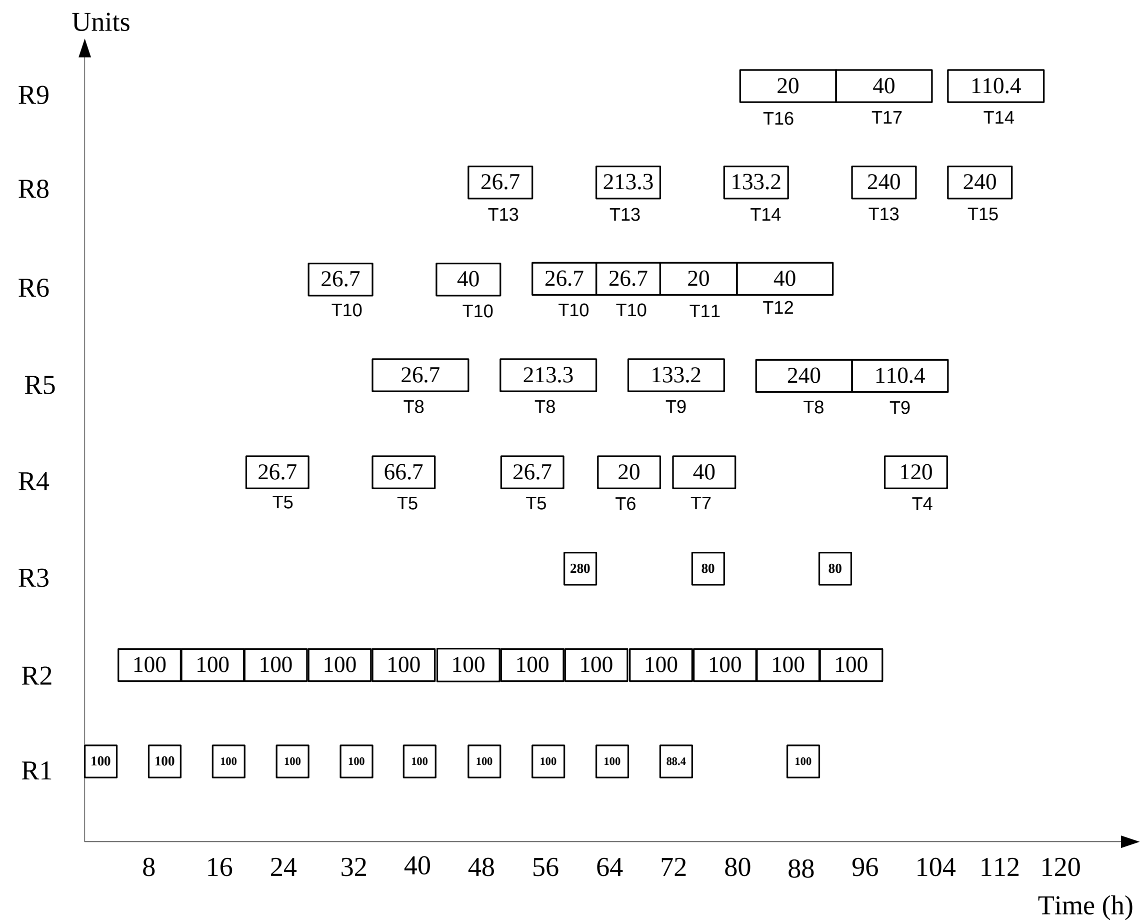


Figure 4.8. Optimal Gantt chart for Case II.

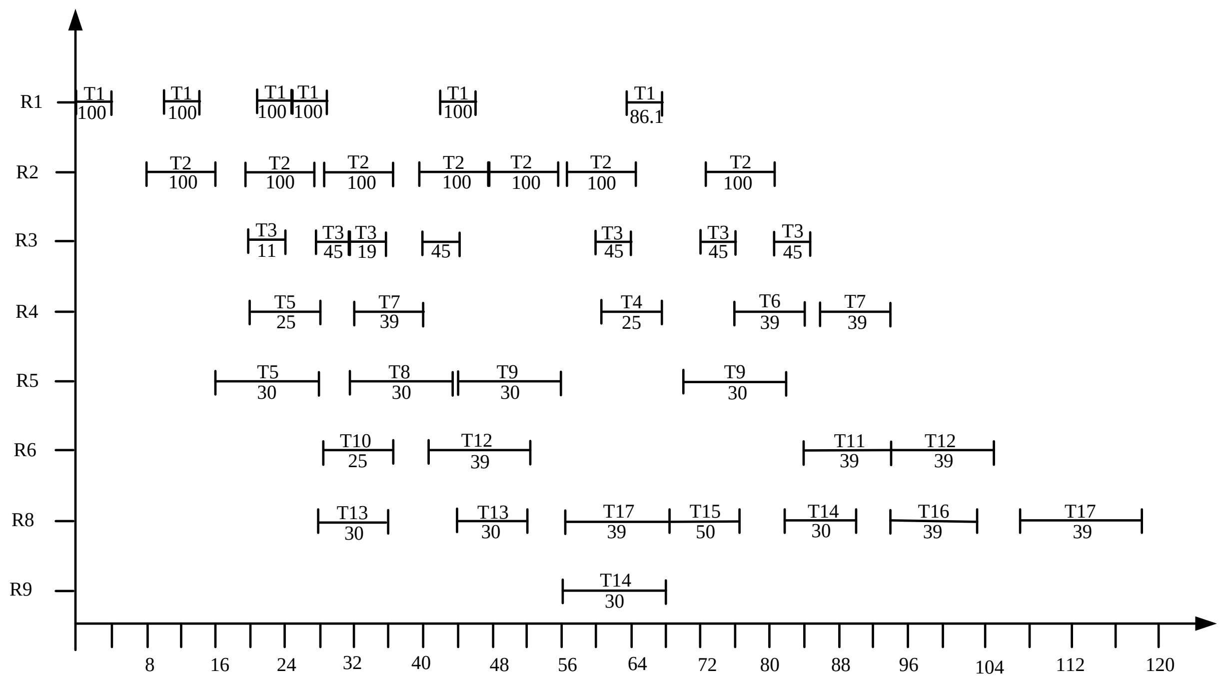


Figure 4.9. Optimal Gantt chart for Case II using the formulation of Pinto et al (2008).

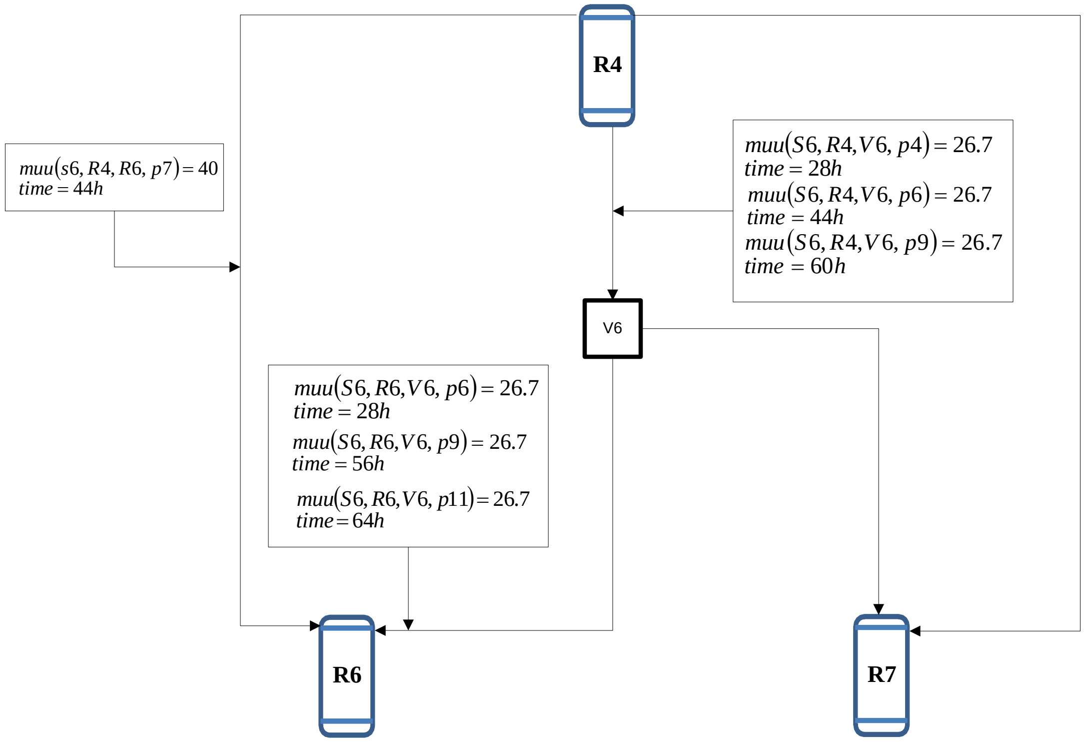


Figure 4.10. Material location in the plant for the state S6.

4.5 Conclusions

A mathematical formulation for the synthesis and design of multipurpose batch plants has been presented. A recent robust scheduling formulation based on the continuous time representation is used as a platform for the design problem and achieves better optimal design and requires less computational time. The objective value obtained in this work improved by 228.6% compared to the recent published formulations for the design and synthesis problem. When this work is compared with other formulations, it gives a smaller size mathematical model that requires fewer binary variables, continuous variables and constraints. The model explicitly considers the different locations of materials in the plant. The formulation also considers the cost arising from the pipe network and determines the optimal pipe network that should exist between equipment.

Nomenclature

Sets

J = { j | j is a piece of equipment}

J_s^{sc} = { j_s^{sc} | j_s^{sc} is a piece of equipment that consumes state s}

J_s^{sp} = { j_s^{sp} | j_s^{sp} is a piece of equipment that produces state s}

V = { v | v is a storage unit}

P = { p | p is a time point}

S_{inJ}^{sc} = { s_{inj}^{sc} | s_{inj}^{sc} is a task which consumes state s}

S_{inJ}^{*} = { s_{inj}^{*} | s_{inj}^{*} is a task performed in unit j}

S_{inJ} = { s_{inj} | s_{inj} is an effective state representing a task}

S_{inJ}^{usc} = { $s_{in,j}^{usc}$ | $s_{in,j}^{usc}$ is a task which consumes unstable state s}

S_{inJ}^{sp} = { s_{inj}^{sp} | s_{inj}^{sp} is a task which produces state s other than a product}

S_{inJ}^{usp} = { s_{inj}^{usp} | s_{inj}^{usp} is a task which produces unstable state s}

S = { s | s is any state}

S^{p} = { s^{p} | s^{p} is any state which is a product}

S_{rm} = { s | s is any state which is a raw material}

S_{inJ}^{p} = { s_{inj}^{p} | s_{inj}^{p} task which produce state s which is a product}

Continuous variables

$tu(s_{inj}, p)$= time at which a task in unit j starts

$tp(s_{inj}, p)$ = time at which a task in unit j finishes

$ss(j)$= design capacity of unit j

$s(v)$= design capacity of storage unit v

$mu(s_{inj}, p)$ = amount of material processed by a task

$muu(s, j, p)$ = amount of state s transferred between unit j and storage unit at time point p.

$mux(s, j, j', p)$ = amount of state *s* transferred between unit *j* and another unit *j*' at time point *p*.

$q_s(s, p)$= amount of state *s* stored at time point *p* in storage unit

$pip(j, v)$ = design capacity of a pipe that connects unit *j* and storage unit *v*

$pipj(j, j')$ = design capacity of a pipe that connects unit *j* and another unit *j''*

Binary variables

$y(s_{inj}, p)$ = 1 if state *s* is used in unit *j* at time point *p*; 0 otherwise

$e(j)$ = 1 if unit *j* is selected; 0 otherwise

$eu(v)$ = 1 if storage unit *v* is selected; 0 otherwise

$z(j, v)$ = 1 if pipe that connects unit *j* and storage unit *v* is selected; 0 otherwise

$w(j, j')$ = 1 if pipe that connects unit *j* and *j'* is selected; 0 otherwise

$t(j, p)$ = 1 if the state produced by unit *j* at time point *p* is consumed; 0 otherwise.

Parameters

M = any large number

$\tau(s_{inj})$ = duration of a task conducted in unit *j*

$\beta(s_{inj})$.= coefficient of variable term of processing time of a task

V_j^L = lower bound for unit *j*

V_j^U = upper bound for unit *j*

V_v^L = lower bound for storage unit *v*

V_v^U = upper bound for storage unit *v*

$\gamma_{s_{inj}}^L$ minimum percentage equipment utilization for a task

$\gamma_{s_{inj}}^U$ maximum percentage equipment utilization for a task

$QO(s)$.= initial amount of state *s* stored in unit

$\rho(s_{inj}^{sp})$ = portion of state *s* produced by a task

$\rho(s_{inj}^{sc})$ = portion of state *s* consumed by a task

$V^{L}_{j,v}$ = lower bound for the capacity of the pipe connecting unit *j* and storage unit *v*

$V^{U}_{j,v}$ = upper bound for the capacity of the pipe connecting unit *j* and storage unit *v*

$V^{L}_{j,j'}$ = lower bound for the capacity of the pipe connecting unit *j* and another unit *j''*

$V^{U}_{j,j'}$ = upper bound for the capacity of the pipe connecting unit *j* and another unit *j''*

$price(s^{p})$ = selling price for a product

AWH = annual working hour

H = time horizon of interest

CRM = cost of raw material

FOC = fixed operating cost for a task

VOC = variable operating cost for a task

FEC = fixed capital cost of equipment

VEQ = variable capital cost of equipment

$FECS$ = fixed capital cost of storage

$VEQS$ = variable capital cost of storage

$FCNC$ = fixed capital cost for pipe connection between processing units

$VCNC$ = variable capital cost for pipe connection between processing units

CNC = fixed capital cost for pipe connection between processing unit and storage

VCN = variable capital cost for pipe connection between processing unit and storage

References

Barbosa-Povoa, A.P. Mateus, R., Novais, A.Q., 2002. Optimal design and layout of industrial facilities: An application to multipurpose batch plants. *Industrial & Engineering Chemistry Research*. 41(15), 3610–3620.

Barbosa-Povoa, A.P., Macchietto, S., 1994. Detailed design of multipurpose batch plants. *Computers & Chemical Engineering*. 18(11/12), 1013–1042.

Barbosa-Povoa, A.P., Mateus, R., Novais, A.Q., 2001. Optimal 2D design layout of industrial facilities. *International Journal of Production Research*. 39(12), 2567–2593.

Barbosa-Povoa, A.P., Pantelides, C.C., 1997. Design of multipurpose plants using the resource-task network unified framework. *Computers & Chemical Engineering*. 21, S703–S708.

Barbosa-Povoa, A.P.F.D., 2007. A critical review on the design and retrofit of batch plants. *Computers & Chemical Engineering*. 31 (7), 833–855.

Castro, P.M., Barbosa-Povoa, A.P., Novais, A.Q., 2005. Simultaneous design and scheduling of multipurpose plants using resource task network based continuous-time formulations. *Industrial & Engineering Chemistry Research*. 44(2), 343–357.

Cerda, J., Vicente, M., Gutierrez, J.M., Esplugas, S., Mata, J., 1989. A new methodology for the optimal-design and production schedule of multipurpose batch plants. *Industrial & Engineering Chemistry Research*. 28(7), 988–998.

Faqir, N.M., Karimi, I.A., 1989. Optimal-design of batch plants with single production routes. *Industrial & Engineering Chemistry Research*. 1989, 28(8), 1191–1202.

Faqir, N.M., Karimi, L.A., 1990. Optimal design of batch plants with multiple production routes. *Foundations of Computer Aided Process Design*. 451–468.

Fuchino, T., Muraki, M., Hayakawa, T., 1994. Scheduling method in design of multipurpose batch plants with constrained resources. *Journal of Chemical Engineering of Japan*. 27(3), 363-368.

Georgiadis, M.C., Rotstein, G.E., Macchietto, S., 1997. Optimal layout design in multipurpose batch plants. *Industrial & Engineering Chemistry Research*. 36(11), 4852–4863.

Heo, S.K., Lee, K.H., Lee, H.K., Lee, I.B., Park, J.H., 2003. A new algorithm for cyclic scheduling and design of multipurpose batch plants. *Industrial & Engineering Chemistry* Research. 42(4), 836–846.

Ierapetritou, M.G., Floudas, C.A., 1998. Effective continuous-time formulation for short-term scheduling: 1. Multipurpose batch processes. *Industrial and Engineering Chemistry Research*. 37, 4341-4359.

Imai, M.; & Nishida, N., 1998. New procedure generating suboptimal configurations to the optimal-design of multipurpose batch plants. *Industrial & Engineering Chemistry Process Design and Development*. 23(4), 845–847.

Kallrath, J.,2002. Planning and Scheduling in the Process Industry. *OR Spectrum*. 24 (3), 219–250.

Klossner, J., & Rippin, D.W.T., 1984. Combinatorial problems in the design of multiproduct batch plants—Extension to multiplant. and partly parallel operation. In *AIChE Annual Meeting*.

Kondili, E.,Pantelides, C.C.,& Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations–1. Mixed integer linear programming formulation. *Computers & Chemical Engineering*17, 211–227.

Lin, X., Floudas, C.A., 2001. Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. *Computers & Chemical Engineering*. 25665–674.

Pantelides, C.C., 1994. Unified frameworks for the optimal process planning and scheduling. *Foundations on Computer Aided Process Design*. 253.

Papageorgaki, S., Reklaitis, G.V., 1990. Optimal design of multipurpose batch plants. 1. Problem formulation. *Industrial & Engineering Chemistry Research. 29*(10), 2054–2062.

Patziatsis, D. I., Xu, G., Papageorgiou, L.G., 2005. Layout Aspects of pipeless batch plants. *Industrial & Engineering Chemistry Research*. 44(15), 5672–5679.

Penteado, F.D., Ciric, A.R., 1996. An MINLP approach for safe process plant layout. *Industrial. & Engineering. Chemistry. Research*. 35, 1354–1361.

Pinto, T., Barbosa-Povoa, A. P., Novais, A. Q., 2008. Design of multipurpose batch plants: A comparative analysis between the STN, m-STN, and RTN representations and formulations. *Industrial & Engineering Chemistry Research*. 47, 6025-6044.

Seid, E.R., Majozi, T., 2012. A robust mathematical formulation for multipurpose batch plants. *Chemical Engineering Science*. 68:36–53.

Shah, N., Pantelides, C.C., 1991. Optimal long-term campaign planning and design of batch-operations. *Industrial & Engineering Chemistry Research*. 30(10), 2308–2321.

Suhami, I., & Mah, R. S. H., 1982. Optimal design of multipurpose batch plants. *Industrial & Engineering Chemistry Process Design and Development*. *21*(1), 94–100.

Vaselenak, J.A., Grossmann, I.E., Westerberg, A.W., 1987. An embedding formulation for the optimal scheduling and design of multipurpose batch plants. *Industrial & Engineering Chemistry Research*. *26*(1), 139–148.

Voudouris, V.T., Grossmann, I.E., 1996. MILP model for scheduling and design of a special class of multipurpose batch plants. *Computers & Chemical Engineering*. 20(11), 1335–1360.

Xia, Q.S., Macchietto, S., 1997. Design and synthesis of batch plants MINLP solution based on a stochastic method. *Computers & Chemical Engineering*. 21, S697–S702.

5

CHAPTER 5

Medium-Term Scheduling of a Multiproduct Industrial Batch Plant

Abstract

A medium-term scheduling technique for a multiproduct batch plant was developed. The intractability of the short-term scheduling models when directly applied to the medium-term scheduling problems was solved by applying a decomposition method. The decomposition model includes a two level mathematical model. The first level determines the type of products and their amount to be produced in each scheduling subproblem to satisfy the market requirement. The second level determines the detailed sequencing of tasks for the tractable size of the subproblems. The recent robust short-term scheduling model based on continuous time (Seid and Majozi, 2012) was extended for solving the scheduling subproblems of the second level decomposition model. The model was applied for solving the medium-term scheduling problem of a pharmaceutical industry specialized in animal vaccines using the actual data. The model effectively solved a makespan minimization problem for the medium-term scheduling horizon of almost 13 weeks.

Keywords: Medium-term scheduling; Multiproduct batch plant; Optimization; MILP.

5.1 Introduction

The pharmaceutical industry is one of the batch processing industries which come into a very competitive environment where customers constantly seek low price as well as high service levels and flexibility. This industry allows flexible multi–product production processes to help companies to respond to changing customer demand and increase plant utilization, but the greater complexity of these processes together with the altered market conditions have caused planning and scheduling techniques to be a challenging task. A considerable amount of attention to the problem of production scheduling and planning for multiproduct and multipurpose batch plants has been given during the last two decades by academic researchers and people in industry.

Extensive reviews have been written by Reklaitis (1992), Shah (1998), Floudas and Lin (2004), Méndez et al. (2006) and Shaik et al. (2006). Most of the work in the area of multiproduct batch plants has dealt with either the long-term planning problem or the short-term scheduling problem. Both planning and scheduling deal with the allocation of available resources over time to perform a set of tasks required to manufacture one or more products. However, long-term planning problems deal with longer time horizons (e.g., several months or years) and are focused on higher-level decisions such as timing and location of additional facilities and levels of production. In contrast, short-term scheduling models address shorter time horizons (e.g., several days) and are focused on determining detailed sequencing of various operational tasks. The area of medium-term scheduling, however, which involves medium time horizons (e.g., several weeks) and still aims to determine detailed production schedules, can result in very large-scale problems and has received much less attention in the literature (Janak and Floudas, 2006; Verderame and Floudas, 2008).

Medium-term scheduling problems can be computationally expensive and these problems are solved using mathematical techniques. The most widely employed strategy to overcome the computational difficulty is based on the idea of decomposition. The decomposition approach divides a large and complex problem, which may be computationally expensive or even intractable when formulated and solved directly as a single MILP model, to smaller subproblems, which can be solved much more efficiently. There have been a wide variety of decomposition approaches proposed in the literature. In addition to decomposition techniques developed for general forms of MILP problems, various approaches that exploit the characteristics of specific process scheduling problems have also been proposed.

In most cases, the decomposition approaches only lead to suboptimal solutions; however, they substantially reduce the problem complexity and the solution time, making MILP-based techniques applicable for large, real-world problems. Pinto and Grossmann (1995) proposed a decomposition scheme for the scheduling of multistage batch plants that may contain equipment in parallel. The solution strategy employed first solves a MILP model which minimizes the total in-process time while determining the assignments of orders to units. Next, an LP model is solved to minimize earliness and eliminate unnecessary setups. Wilkinson et al. (1995) presented a method for producing accurate aggregate models of multipurpose plant operations which represents a strict relaxation of the detailed scheduling model. The aggregate formulation provides a tight upper bound on the solution to the original problem and can be solved in considerably less computational time.

Bassett et al. (1996) discussed a number of decomposition-based approaches for the solution of large-scale batch scheduling problems. The first approach utilizes a time-based decomposition with a hierarchical approach that separates the problem into a planning level and a scheduling level. These levels are solved iteratively, and various techniques are used to remove infeasibilities. A second approach, called a reverse rolling window, utilizes a hybrid planning/scheduling formulation in which only a small section of the horizon is determined in detail at each iteration. Then, sequences of such problems are solved in reverse order for the time instances. The authors also consider a resource-based decomposition approach and a task-unit aggregation approach. Dimitriadis et al. (1997) proposed forward and backward rolling-horizon algorithms for the solution of medium-term scheduling problems. These rolling-horizon algorithms are based on separating the overall scheduling horizon into two blocks of time, a detailed time block and an aggregate time block. The problem is then solved in a sequence of iterations where, at each iteration, the previous solution of the detailed time block is fixed and part of the aggregate time block is moved into the detailed time block.

Elkamel et al. (1997) developed a heuristic decomposition algorithm for the scheduling of batch plants which consists of two basic components. First, longitudinal decomposition is used to partition the units in the plant into different subsets capable of performing different sets of tasks. Next, axial decomposition is used to assign the product orders into groups based on their due dates as well as the unit groups. Then, the orders are scheduled sequentially along the time axis. Gupta and Maranas (1999) proposed a procedure to partition a large, mid-term planning problem into smaller subproblems using hierarchical Lagrangean relaxation of key complicating constraints. This procedure is used together with an upper-bound generating heuristic within a subgradient optimization framework. Harjunkoski and Grossmann (2001) presented a decomposition strategy for the scheduling of production in an industrial steel plant. First, customer orders are partitioned into groups with similar properties, and each group is optimally scheduled as a flowshop problem. Then, a LP/MILP model is used to account for setup times and properly allocate some shared equipment.

Lin et al. (2002) considered the medium-range production scheduling of a multiproduct batch plant using a rolling-horizon approach. In their approach, the overall scheduling problem is decomposed into a series of smaller short-term scheduling subproblems in successive time horizons, which are connected through material and unit availabilities. A two-level decomposition framework is utilized to determine the current time horizon and the products to be included, which takes into account demand distribution and unit utilization and imposes limits on the complexity of the resulting short-term scheduling problem. Then, for each subhorizon, a continuous-time, MILP-based short-term scheduling model is applied to determine the detailed production schedule. The decomposition model and the short-term scheduling model are solved iteratively for each short horizon until the schedules for the whole period under consideration are generated.

Wu and Ierapetritou (2003) presented a number of different heuristic-based decomposition approaches for the efficient solution of large-scale scheduling problems. They consider time based decomposition, required production method, and resource based decomposition. Their formulation utilizes Lagrangean relaxation and Lagrangean decomposition to generate upper bounds that are used in an iterative fashion with the lower bounds obtained through the heuristic approaches to obtain production schedules for realistic sized problems.

Janak and Floudas (2006) proposed an enhanced state task network MILP model for the medium-term production scheduling of a multiproduct industrial batch plant. The proposed approach extends the work of Ierapetritou and Floudas (1998) and Lin et al. (2002) to consider a large-scale production facility and account for various storage policies (UIS, NIS and ZW), variable batch sizes and processing times, batch mixing and splitting, sequence dependent changeover times, intermediate due dates, products used as raw materials and several modes of operation.

In this work, a mathematical model for the medium-term scheduling of a pharmaceutical industry is proposed based on real industrial data. The proposed model decomposes the large and complex problem into smaller short-term scheduling subproblems in successive time horizons. The model handles the different storage policies: fixed intermediate storage, no intermediate storage, multipurpose storage where different states can be stored in a storage vessel, a state can be stored in different available storage, a state has to wait for a certain duration before it can be used in the upper production stream. These features make the proposed model different from other formulations where there is dedicated storage for a state and storage dedicated for a state. Additionally, the recent short-term scheduling technique by Seid and Majozi (2012) is enhanced to solve the short-term scheduling part after decomposing the large medium-term scheduling problem into smaller short-term scheduling problems. The rest of this chapter is organized as follows. In Section 5.2, the problem statement is presented followed by the mathematical formulation presented in Section 5.3. The industrial case study and its solution are provided in Section 5.4. Finally a conclusion is drawn in Section 5.5.

5.2 Problem statement

The problem statement for the medium-term production scheduling problem of a multiproduct batch plant is defined as follows:

Given

(i) the production recipe (i.e., the processing times for each task in the suitable units and the amounts of materials required for the production of each product)

(ii) the available units and their capacity limits

(iii) the available storage capacity for each of the materials and

(iv) the medium-range time horizon under consideration, then the objective is to **determine**

a) the optimal sequence of tasks taking place in each unit

b) the amount of material being processed at each time in each unit and

c) the processing time of each task in each unit, so as to satisfy the market requirements expressed as specific amounts of products at given time instances within the time horizon while maximizing the production or minimizing the makespan.

5.3 Overall decomposition model

The overall methodology for solving the medium-range production scheduling problem of a pharmaceutical industry considered in this work is to decompose the large and complex problem into smaller short-term scheduling subproblems in successive time periods. The first level is solving a simple MILP model to determine the number of products and amount of a product that should be produced in each decomposed subproblem by considering the medium-range time horizon. The objective of the decomposition is to minimize the number of events evolved in each subproblem so that a moderate sized problem can be achieved that can be solved with a reasonable computational effort. The second level is a short-term scheduling subproblem that contains two steps. The first step is solving the subproblem as a makespan minimization to achieve the amount that should be produced for each product determined by the first level of the decomposition problem. The second step is adding additional products from the next period and maximizing the production of the newly added products while using the time horizon determined by the first step and fixing the amount that should be produced for each product involved in the selected time period. Consequently, the second step maximizes the utilization of the units of the current period. The first and second steps in the second level are solved successively until all periods are finished.

5.3.1 Mathematical model for the first level.

The first level of the decomposition model is a relaxed MILP scheduling problem where detailed sequencing of tasks is avoided. The model involves the following constraints:

Capacity constraint

Constraint (5-1) implies that the total amount of material consumed by task *i* at time period *k* is limited by the capacity of the unit.

$$b(i,j,k) \le V_j^U n(i,j,k), \ \forall \ k \in K, \ j \in J, \ i \in I \qquad (5\text{-}1)$$

Material balance for storage

Constraint (5-2) states that the amount of material stored at each time period *k* is the amount stored at the previous time period adjusted to some amount resulting from the difference between state *s* produced by the producing tasks and the consuming tasks at the current period *k*. This constraint is used for a state other than a product, since the latter is not consumed, but only produced within the process.

$$q_s(s,k) = q_s(s,k-1) - \sum_{i \in I^{sc}} \sum_{j \in J^{sc}} \rho_i^{sc} b(i,j,k) + \sum_{i \in I^{sp}} \sum_{j \in J^{sp}} \rho_i^{sp} b(i,j,k), \quad \forall \ k \in K, \ s \in S \qquad (5\text{-}2)$$

Constraint (5-3) states that the amount of product stored at time period *k* is the amount stored at the previous time period *k*-1 and the amount of product produced at time period *k*.

$$q_s(s,k) = q_s(s,k-1) + \sum_{i \in I^{sp}} \sum_{j \in J^{sp}} \rho_i^{sp} b(i,j,k), \ \forall \ k \in K, \ s \in S^p \tag{5-3}$$

Duration constraints (batch time as a function of batch size)

Constraint (5-4) describes the duration constraint modeled as a function of batch size where the processing time is a linear function of the batch size. The finishing time of unit *j* in period *k* should be greater than or equal to the starting time of unit *j* and the processing time of all tasks performed in the unit.

$$t_f(j,k) \ \geq \ t_s(j,k) + \sum_{i \in I_j} \tau(i,j) n(i,j,k) + \beta(i,j) b(i,j,k), \quad \forall \ j \in J, \ k \in K \tag{5-4}$$

Constraint (5-5) defines that the starting time of unit *j* at the current period *k* should be greater than or equal to the time horizon of the previous period. At the beginning, the time horizon takes a value of zero.

$$t_s(j,k) \geq \ h(k-1), \ \forall \ j \in J, \ k \in K \tag{5-5}$$

Demand constraint

Constraint (5-6) describes that the amount of product stored at the final period should satisfy the market requirement.

$$q_s(s,k) \geq Demand(s), \ \forall \ k = K, \ s \ \in S^p \tag{5-6}$$

Time horizon constraint

Constraint (5-7) states that the finishing time of unit *j* should be less than the time horizon of time period *k*.

$$t_f(j,k) \leq h(k), \ \forall \ j \in J, \ k \in K \tag{5-7}$$

Constraint (5-8) states that the finishing time of unit *j* at the final period should be less than or equal to the medium-term time horizon under consideration.

$$t_f(j,k) \geq \ Hmed, \ \forall \ j \in J, \ k = K \tag{5-8}$$

Objective function

The objective function defined for the first level decomposition model is to reduce the total event points in each period to achieve a moderate sized model that can be solved in a reasonable computational time. Constraint (5-9) and (5-10) are used to achieve this objective.

$$n \geq \sum_i \sum_j n(i,j,k), \quad \forall \ j \in J, \ i \in I, \ k \in K \tag{5-9}$$

$$\min n \tag{5-10}$$

5.3.2 Mathematical model for the second level

The mathematical model for the second level of the decomposition technique involves the following constraints.

Allocation constraint

Constraint (5-11) implies that at time point p only one task is allowed to be performed in unit j.

$$\sum_{s_{inj} \in S^*_{inj}} y\left(s_{inj}, p\right) \leq 1, \ \forall \ j \in J, \quad p \in P \tag{5-11}$$

Constraint (5-12) implies that only one type of state s is allowed to be stored in a storage unit u.

$$\sum_{s \in S^u} y(s,u,p) \leq 1, \ \forall \ u \in U, \quad p \in P \tag{5-12}$$

Capacity constraints

Constraint (5-13) implies that the total amount of all the states consumed at time point p is limited by the capacity of the unit which consumes the states. $V^L_{s_{inj}}$ and $V^U_{s_{inj}}$ represent lower and upper bounds in capacity of a given unit that processes the effective state s_{inj}.

$$V^L_{s_{inj}} y\left(s_{inj}, p\right) \leq mu\left(s_{inj}, p\right) \leq V^U_{s_{inj}} y\left(s_{inj}, p\right), \ \forall \ p \in p, \ j \in J, \ s_{inj} \in S_{inJ} \tag{5-13}$$

Constraint (5-14) defines that the amount of state *s* transferred from the producing units to the storage should be limited by the maximum capacity of the storage.

$$\sum_{j \in J} muu(s, j, u, p) \leq V_u^U y(s, u, p) \text{ , } \forall \;\; p \in P, \;\; j \in J, \; u \in U \tag{5-14}$$

Material balance for storage

Constraint (5-15) states that the amount of material stored at each time point *p* is the amount stored at the previous time point adjusted to some amount resulting from the difference between state *s* produced by tasks at the previous time point (*p*-1) and used by tasks at the current time point *p*. This constraint is used for a state other than a product, since the latter is not consumed, but only produced within the process.

$$q_s(s,u,p) = q_s(s,u,p-1) - \sum_{j \in J^{sc}} muu(s,j,u,p) + \sum_{j \in J^{sp}} muu(s,j,u,p-1), \quad \forall \;\; p \in P, \;\; p \geq 1, \;\; s \in S \tag{5-15}$$

Constraint (5-16) is used for material balance around storage at the first time point.

$$q_s(s,u,p) = QO(s,u) - \sum_{j \in J^{sc}} muu(s,j,u,p), \forall \;\; p \in P, \;\; p = 1, \;\; u \in U, \;\; s \in S \tag{5-16}$$

Constraint (5-17) states that the amount of product stored at time point *p* is the amount stored at the previous time point and the amount of product produced at time point *p*.

$$q_s(s,u,p) = q_s(s \;\;,u,p-1) + \sum_{s_{inj} \in \, s_{inJ}^{sp}} \rho_{s_{inj}}^{sp} mu(s_{inj}, p), \; \forall \;\; p \in P, \;\; s^p \in S^p \tag{5-17}$$

Material balance around the processing unit

Constraint (5-18) states that the material processed in the unit equals the amount of material coming directly from the producing units and from the storage.

$$\rho_{s_{inj}}^{sc} mu(s_{inj},p) = \sum_{u \in U^s} muu(s,j,u,p) + \sum_{j' \in J^{sp}} mux(s,j,j',p), \quad \forall \;\; p \in P, \; j \in J^{sc}, \; j' \in J^{sp}, \; s_{inj} \in S_{inJ}, \; s \in S \tag{5-18}$$

Constraint (5-19) is used to define that the amount of material consumed at time point *p* is sent to storage at time point *p* and to units that consume the state at time point *p*+1.

$$\rho_{s_{inj}}^{sp} mu(s_{inj}, p) = \sum_{u \in U^s} muu(s, j, u, p) + \sum_{j' \in J^{sc}} mux(s, j', j, p+1), \quad \forall\ p \in P,\ j \in J,\ s_{inj} \in S_{inJ},\ s \in S \tag{5-19}$$

Duration constraints (batch time as a function of batch size)

Constraint (5-20) describes the duration constraint modeled as a function of batch size where the processing time is a linear function of the batch size. For zero-wait (ZW) only the equality sign is used.

$$t_p(j,p) \geq t_u(j,p) + \sum_{s_{in,j} \in S^*_{in,j}} \left(\tau\left(s_{in,j}\right) y\left(s_{in,j},p\right) + \beta(s_{in,j}) mu(s_{in,j},p)\right), \quad \forall\ j \in J,\ p \in P,\ s_{in,j} \in S_{in,J} \tag{5-20}$$

Constraint (5-21) defines the duration a state should stay in the storage before it is consumed by the consuming units.

$$t_p(u,p) \geq t_u(u,p) + \sum_{s_{in,j} \in S^*_{in,j}} \left(\tau\left(s_{inj}\right) y\left(s_{inj},p\right)\right), \quad \forall\ j \in J,\ p \in P,\ s_{inj} \in S_{inJ} \tag{5-21}$$

Constraint (5-22) and (5-23) state that the starting time of the processing and storage units at the first time point should be greater than or equal to the finishing time of the last batch in the units at the previous time period.

$$t_u(j,p1) \geq ten(j), \forall\ j \in J,\ p \in P \tag{5-22}$$

$$t_u(u,p1) \geq ten(u), \forall\ u \in U,\ p \in P \tag{5-23}$$

Setup time for units and storage units

Constraints (5-24) and (5-25) entails that the setup time of the unit at time point *p* should be greater than the finishing time of the unit at the previous time point *p* plus the setup time if the unit is actively processing a task at time point *p*.

$$t_{sp}(j,p) \geq t_p(j,p) + \sum_{s_{inj} \in S^*_{inj}} \theta(s_{inj})\, y(s_{inj},p), \forall\ j \in J,\ p \in P, s_{inj} \in S_{inJ} \tag{5-24}$$

$$t_{sp}(u,p) \geq t_p(u,p) + \sum_{s \in S^u} \theta(s,u)\, y(s,u,p), \forall\ u \in U,\ p \in P \tag{5-25}$$

Sequence constraints for processing units and storage units

Constraint (5-26) states that the starting time of the processing unit at time point *p* should be equal or later than the setup time of the processing unit at the previous time point *p*-1.

$$t_u(j,p) \geq t_{sp}(j,p-1), \forall\ j \in J,\ p \in P \tag{5-26}$$

Constraint (5-27) states that the starting time of the storage unit at time point *p* should be equal or later than the setup time of the storage unit at the previous time point *p*-1.

$$t_u(u,p) \geq t_{sp}(u,p-1), \forall\ u \in U,\ p \in P \tag{5-27}$$

Sequence constraints relating processing units and storage units

Constraint (5-28) states that if a material is sent from the storage to the processing unit the binary variable $t(u,j,p)$ should take a value of one which makes the sequence constraints (5-29) and (5-30) relating the consuming unit and the storage hold. If no material is sent from storage to the unit, the binary variable $t(u,j,p)$ will take a value of zero to relax the sequence constraints (5-29) and (5-30).

$$\sum_{s \in S^u} muu(s, j, u, p) \leq V_u^U t(u, j, p), \forall\ p \in P,\ j \in J,\ u \in U,\ s \in S \tag{5-28}$$

$$t_u(j,p) \geq t_p(u,p-1) - M(1 - t(u, j, p)), \forall\ u \in U, j \in J^u,\ p \in P \tag{5-29}$$

$$t_u(j,p) \leq t_p(u,p-1) - M(1 - t(u, j, p)), \forall\ u \in U, j \in J^u,\ p \in P \tag{5-30}$$

Sequence constraints relating different processing units

Constraint (5-31) states that if a material is sent from the producing unit to the consuming unit the binary variable *t(j,j',p)* should take a value of one which makes the sequence constraints (5-32) and (5-33) relating the consuming unit and the producing unit hold. If no material is sent from the producing unit to the consuming unit the binary variable $t(j,j',p)$ will take a value of zero to relax the sequence constraints (5-32) and (5-33).

$$\sum_{s \in J^{sp}} muu(s,j,j',p) \le V_{j'}^{U} t(j,j',p), \forall \ p \in P,\ j \in J^{sc}, j' \in J^{sp} \tag{5-31}$$

$$t_u(j,p) \ge t_p(j',p-1) - M(1 - t(j,j'p)), \forall \ j \in J^{sc}, j' \in J^{sp},\ p \in P \tag{5-32}$$

$$t_u(j,p) \le t_p(j',p-1) - M(1 - t(j,j'p)), \forall \ j \in J^{sc}, j' \in J^{sp},\ p \in P \tag{5-33}$$

No batch mixing in the storage

Constraint (5-34) is used to ensure that the amount of state s stored in the storage unit at time point p will take a value of zero if the binary variable related to the storage of another batch of state in the storage unit takes a value of one in order to avoid batch mixing in the storage.

$$q_s(s,u,p) \le V_u^U \left(1 - \sum_{s \in S^u} y(s,u,p)\right), \forall \ p \in P,\ p \ge 1,\ s \in S \tag{5-34}$$

Time horizon constraints

The usage and the production of states should be within the time horizon of interest. These conditions are expressed from Constraints (5-35) to (5-38).

$$t_u(j,p) \le h,\ \forall \ \ p \in P,\ j \in J \tag{5-35}$$

$$t_p(j,p) \le h,\ \forall \ p \in P,\ j \in J \tag{5-36}$$

$$t_u(u,p) \le h,\ \forall \ \ p \in P,\ j \in J \tag{5-37}$$

$$t_p(u,p) \le h,\ \forall \ p \in P,\ j \in J \tag{5-38}$$

Amount of product produced in the scheduling problem

Constraint (5-39) states that the amount of product produced in the scheduling subproblem should satisfy the amount that should be produced obtained from the first level model.

$$q_s(s,p) \geq Demad(s), \ \forall \ p = P, \ s \in S^p \tag{5-39}$$

Objective function

The objective function presented in Constraint (5-40) is the minimization of makespan.

$$\min h \tag{5-40}$$

The first step of the second level model contains Constraints (5-11) to (5-40) and it is the makespan minimization problem obtained by setting the amount of product that should be produced in the subproblem determined by the first level decomposition model.

The second step of the second level model contains Constraints from (5-11) to (5-39) and Constraint (5-41) and it is a maximization problem that maximizes production by including products from the next cycle. At this stage the time horizon obtained from the first step is used. This step allows the utilization of the units to produce the product taken from the next period if there are enough available resources and should occur within the time horizon obtained by the first step. Constraint (5-41) also minimizes the finishing time of units so that it will give the units a chance in the next time period to start earlier that the makespan of the current time period.

Objective function for the second step of second level model

$$\max \sum_{s \in S^p} q_s(s,p) - \left(\sum_j t_p(j,p) - \sum_u t_p(u,p) \right), \ \forall \ p = P, \ s \in S^p, \ S^{pnk} \tag{5-41}$$

5.4 Case study

The case study is taken from one of the pharmaceutical industries in South Africa. The production processes follow a multiproduct nature where all the products follow the same production path. The state task network representation for this plant is presented in Figure 5.1. It is a complex problem which involves 36 tasks, 48 states and 12 raw materials. The product passes through four different tasks. The second task is a storage task where the state should stay for a while for quality inspection before it is used by task 3. The scheduling data related to this case is presented in Table 5.1 and 5.2. The due date for all products is assumed at the end of 180 days.

Table 5.1. Scheduling data for the industrial case study.

Task	Suitable units	$\tau(s_{in,j})$day	$\theta(s_{in,j})$day	$V^L_{s_{in,j}} - V^U_{s_{in,j}}$
Task (T1-T11)	U1	0.32	0.17	0-238
Task (T1-T11)	U2	0.32	0.17	0-24
Task (T12-T22)	U3-U5	12.1	0.17	0-238
Task (T12-T22)	U6-U10	12.1	0.14	0-238
Task (T23-T29)	U11 and U12	0.364	0.25	0-2722
Task (T30-T36)	U13	0.21	0.25	0-2722

Table 5.2. Material data for the industrial case study.

States	Storsge capacity (mu)	Initial inventory	Demand requirment
S1-S11	UL	AA	0
S12-S33	NV	0	0
S34,S36,S40,S42,S44,S46	NV	0	0
Product 1	UL	0	902.5
Product 2	UL	0	5301.5
Product 3	UL	0	2610
Product 4	UL	0	1943
Product 5	UL	0	4585
Product 6	UL	0	7899
Product 7	UL	0	5129

UL = unlimited; NV = not available; AA = available as required

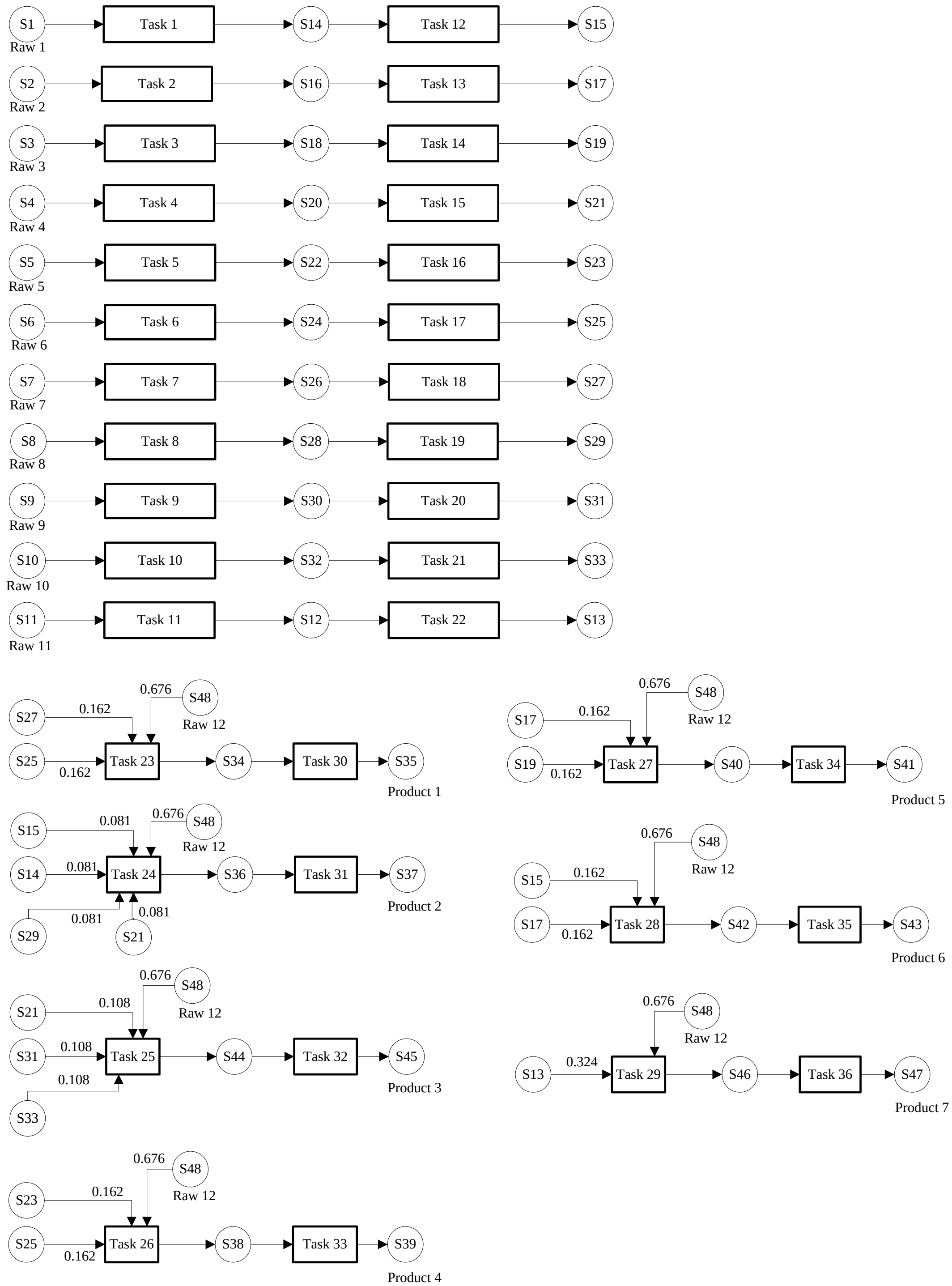


Figure 5.1 STN representations for the case study.

Results and discussion

The results for the case study using the proposed model were obtained using CPLEX 9.1.2/ GAMS 22.0 on a 2.4 GHz, 4 GB of RAM, Acer TravelMate 5740G computer. The first level decomposed model was solved to determine the type of products and the amount that should be produced in order to meet the demand requirement. The results obtained are presented in Table 5.3. For the first time period (k1) the model chose product 1, product 2 and product 4, with respective batch sizes of 905.2 m.u., 2722 m.u. and 566.6 m.u. for production. For time period (k2) product 4, product 5 and product 6 are selected with respective amount of 1469 m.u., 2722 m.u. and 2722 m.u., For the rest of the time periods the selected products and the amounts associated with the products are presented in Table 5.3. The distribution of the products to time periods gives a controlled tractable size of the short-term scheduling subproblem. For the time period of more than 5 the model size of the subproblems does not reduce considerably. Consequently, a time period of 5 was chosen to solve the second level decomposition of the short-term scheduling problem. The model contains 2520 integer variables, 3567 continuous variables 1433 constraints and solved in a specified CPU time of 10 minutes.

Table 5.3. Computational results for the first level decomposition model for the case study.

Products	Time period K1	Time period K2	Time period K3	Time period K4	Time period K5
product 1	905.2	0	0	0	0
product 2	2722	0	0	2579.5	0
product 3	0	0	2610	0	0
product 4	566.6	1469	0	0	0
product 5	0	2722	1863	0	0
product 6	0	2722	1075.3	4890	0
product 7	0	0	0	0	5129

The results for each time period for the short-term scheduling subproblems is shown in Table 5.4. For the first period, a makespan minimization of 16.12 days was obtained with the satisfaction of the demand imposed by the first level decomposition model (902 m.u. for Product 1, 2722 m.u. for product 2 and 863 m.u. for product 4). The model contains 950 binary variables, 5022 continuous variables, 5045 constraints and solved in a specified CPU time of 1 h. The finishing time of the last batch in the processing units in the first period was set as the initial starting time for the second period to solve the makespan minimization. An objective value of 30.27 days was obtained. The solution continued the same way until the all the time periods were finished. The makespan for the final period was 95.7 days which is also the total number of days required to produce all the products with the satisfaction of the market demand. The decomposition model solved the makespan minimization of the medium range scheduling problem for the case study, which is difficult to solve when the short-term scheduling model alone is implemented. The Gantt chart for the first period is depicted in Figure 5.2.

Table 5.4. Computational results for the second level for the case study.

Time periods	Time points	Binary variables	Continuous variables	Constraints	Time horizon	CPU (h)
K1	10	950	5022	5045	16.12	1
K2	12	876	4586	5011	30.27	1
K3	10	840	4422	4618	56.03	1
K4	13	767	5280	4827	82.47	1
K5	13	299	1458	3006	95.7	1

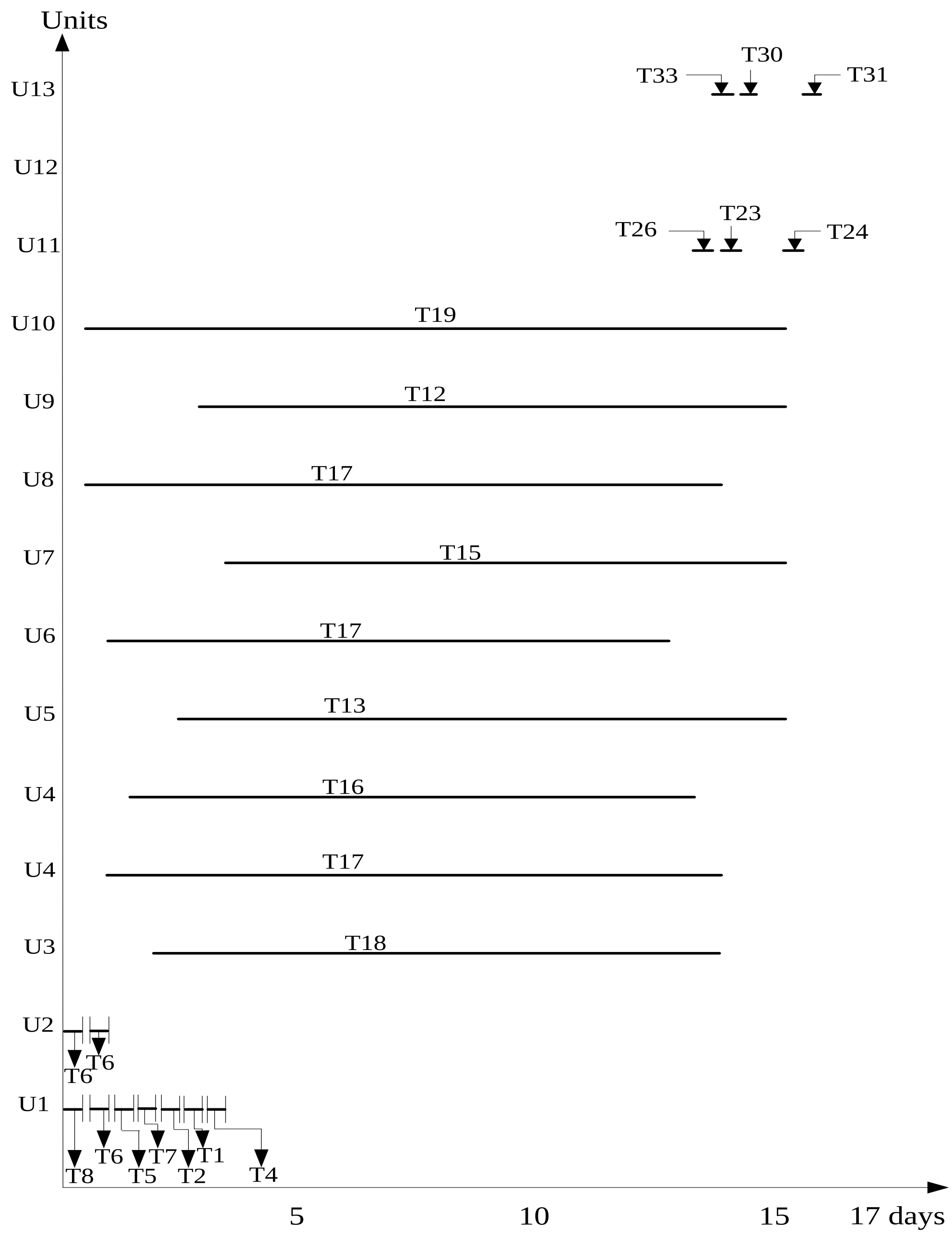


Figure 5.2 Gantt chart for the first period for the Case study.

5.5 Conclusions

In this work, a medium-term production scheduling technique for multiproduct batch plants is presented. The developed model is tested with a real case from the pharmaceutical industry. The solution technique for the problem is by the use of a decomposition approach where the large scale problem is divided into a tractable sized short-term scheduling subproblem. The decomposition model was implemented to solve a makespan minimization for almost 13 weeks, which is difficult to solve when the short-term scheduling model is applied for this medium-term time horizon. Additional features of the model are the incorporation of the following unique characteristics of a storage vessel: storage can be used for different states, a state can be stored in multiple storage units, a state should wait in storage for certain duration before it is used and no batch mixing in storage is permitted. The proposed model can be easily extended to incorporate intermediate due dates for a product.

Nomenclature

Level 1 formulation

Sets

J = { j | j is a unit}

I = { i | i is a task}

K = { k | k is a time period}

I^{sc} = { i^{sc} | i^{sc} is a task that consumes state s}

J^{sc} = { j^{sc} | j^{sc} is a unit that consumes state s}

I^{sp} = { i^{sp} | i^{sp} is a task that produces state s}

J^{sp} = { j^{sp} | j^{sp} is a unit that produces state s}

J^{u} = { j^{u} | j^{u} is a unit that accepts material from storage unit u}

S = { s | s is a state}

S^{p} = { s | s is a state which is a product}

I_j = { $i_j | i_j$ is a task performed in unit *j*}

Parameters

V_j^U = upper bound for unit *j*

ρ_i^{sp} = portion of state *s* produced by a task

ρ_i^{sc} = portion of state *s* consumed by a task

$\tau(i, j)$ = duration of the task conducted in unit *j*

$\beta(i, j)$.= coefficient of variable term of processing time of a task conducted in unit *j*

$Demand(s)$.= market demand for state *s*

H_{med} .= time horizon for the medium term scheduling problem

Variables

$b(i, j, k)$.= amount of material processed by the task in unit *j* at time period *k*

$q_s(s, k)$.= amount of state *s* stored at period *k*

$t_f(j, k)$.= finishing time of unit *j* in period *k*

$t_s(j, k)$.= starting time of unit *j* in period *k*

$h(k)$.= time horizon for period *k*

n .= maximum number of event points for the scheduling subproblem

Binary variable

$n(i, j, k)$.= Integer variable for number of batches of task *i* in unit *j* at time period *k*

Level 2 formulation

Sets

J = { $j | j$ is a unit}

U = { $u | u$ is a storage}

P = { $p | p$ is a time point}

S = { $s | s$ is any state}

S^u = { $s^u | s^u$ is any state stored in storage unit u}

S_{inj}^* = { $s_{inj}^* | s_{inj}^*$ tasks performed in unit *j*}

S_{inJ} = { s_{inj} | s_{inj} is an effective state representing a task}

S_{inJ}^{sp} = { s_{inj}^{sp} | s_{inj}^{sp} task which produce product state}

J^{sc} = { j^{sc} | j^{sc} is a unit that consumes state s}

J^{sp} = { j^{sp} | j^{sp} is a unit that produces state s}

S^{p} = { s | s is any state which is a product}

U^{s} = { u^{s} | u^{s} is any storage used to store state s}

S^{pnk} = { s^{pnk} | s^{pnk} is any state which is a product from the next time period}

Parameters

V_{j}^{U} = upper bound for unit j

V_{u}^{U} = upper bound for storage unit u

$V_{s_{inj}}^{U}$ = maximum capacity of unit j to process a particular task

$V_{s_{inj}}^{L}$ = minimum capacity of unit j to process a particular task

$QO(s,u)$ = initial available amount of state s in storage unit u

$\rho_{s_{inj}}^{sp}$ = portion of state s produced by a task

$\rho_{s_{inj}}^{sc}$ = portion of state s consumed by a task

$\tau(s_{inj})$ = constant coefficient of processing time of a task

$\beta(s_{inj})$.= coefficient of variable term of processing time of a task

$\tau(s,u)$ = duration of state s stored in unit u before it is consumed by the consuming task

$\theta(s_{inj})$ = setup time of unit j after performing a task

$\theta(s,u)$ = setup time of storage unit u after storing state s

M = big number equivalent to the medium-range time horizon

$Demand(s)$.= market demand for state s

Variables

$mu(s_{inj}, p)$ = amount of material processed by the task

$muu(s, j, u, p)$ = amount of sate *s* transferred between unit *j* and storage unit *u* at time point *p*.

$mux(s, j, j', p)$ = amount of sate *s* transferred between unit *j* and another unit *j'* at time point *p*.

$q_s(s, u, p)$= amount of state *s* stored at time point *p* in storage unit *u*

$t_p(j, p)$.= finishing time of unit *j* at time point *p*

$t_u(j, p)$= starting time of unit *j* at time point *p*

$t_p(u, p)$.= finishing time of storage unit *u* at time point *p*

$t_u(u, p)$= starting time of storage unit *u* at time point *p*

$t_{en}(j, k)$.= finishing time of the last batch in unit *j* at time period *k*

$t_{en}(u, k)$.= finishing time of the last batch stored in storage unit *u* at time period *k*

$t_{sp}(j, p)$.= setup time for unit *j* at time point *p*

$t_{sp}(u, p)$.= setup time of storage unit *u* at time point *p*

Binary variables

$y(s_{inj}, p)$ = 1 if state *s* is used in unit *j* at time point *p*; 0 otherwise

$y(s, u, p)$ = 1 if state *s* is stored in storage unit *u* at time point *p*; 0 otherwise

$t(u, j, p)$ = 1 if the state used in unit *j* is from storage unit *u* at time *p*; 0 otherwise.

$t(j, j', p)$ = 1 if the state used in unit *j* is from storage unit *j'* at time *p*; 0 otherwise.

References

Bassett, M.H., Pekny, J.F., Reklaitis, G.V., 1996. Decomposition Techniques for the Solution of Large-Scale Scheduling Problems. *AIChE J.* 42, 3373.

Dimitriadis, A.D., Shah, N., Pantelides, C.C., 1997. RTN-Based Rolling Horizon Algorithms for Medium Term Scheduling of Multipurpose Plants. *Comput. Chem. Eng*. 21, S1061.

Elkamel, A., Zentner, M., Pekny, J.F., Reklaitis, G.V., 1997. A Decomposition Heuristic for Scheduling the General Batch Chemical Plant. *Eng. Optim*. 28, 299.

Floudas, C.A., Lin, X., 2004. Continuous-Time versus Discrete-Time Approaches for Scheduling of Chemical Processes: A Review. *Comput.Chem. Eng*. 28, 2109.

Gupta, A., Maranas, C.D., 1999. A Hierarchical Lagrangean Relaxation Procedure for Solving Midterm Planning Problems. *Ind. Eng. Chem. Res*. 38, 1937.

Harjunkoski, I., Grossmann, I.E., 2001. A Decomposition Approach for the Scheduling of a Steel Plant Production. *Comput. Chem. Eng*. 25, 1647.

Ierapetritou, M.G., Floudas, C.A., 1998. Effective Continuous-Time Formulation for Short- Term Scheduling. 2. Continuous and Semicontinuous Processes. *Ind. Eng. Chem. Res*. 37, 4360.

Janak, S.L., Floudas, C.A., Vormbrock, N., 2006. Production Scheduling of a Large-Scale Industrial Batch Plant. I. Short-Term and Medium-Term Scheduling. *Ind. Eng. Chem. Res*. 25, 8234.

Lin, X., Floudas, C.A., Modi, S., Juhasz, N.M., 2002. Continuous-Time Optimization Approach for Medium-Range Production Scheduling of a Multiproduct Batch Plant. *Ind. Eng. Chem. Res*. 41, 3884.

Me´ndez, C.A., Cerda´, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. State of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng*. 30, 913–946.

Pinto, J.M., Grossmann, I.E., 1995. A Continuous Time Mixed Integer Linear Programming Model for Short-Term Scheduling of Multistage Batch Plants. *Ind. Eng. Chem. Res*. 34, 3037.

Reklaitis, G.V., 1992. Overview of Scheduling and Planning of Batch Process Operations. Presented at NATO Advanced Study Institutes Batch Process Systems Engineering, Antalya, Turkey.

Seid, R., Majozi, T., 2012. A robust mathematical formulation for multipurpose batch plants. *Chem. Eng. Sci.* 68:36–53.

Shah, N., 1998. Single- And Multisite Planning and Scheduling: Current Status and Future Challenges. In Proceedings of the Third International Conference on Foundations of Computer-Aided Process Operations, *AIChE, Snowbird, UT, Pekny, J. F., Blau, G. E., Eds., CACHE: New York*. pp 75-90.

Shaik, M.A., Janak, S.L., Floudas, C.A., 2006. Continuous-time models for short-term scheduling of multipurpose batch plants: a comparative study. *Ind. Eng. Chem. Res*. *45*, 6190–6209.

Verderame, P.M., Floudas, C.A., 2008. Integrated operational planning and medium-term scheduling of a large-scale industrial batch plants. *Ind. and Eng. Chem. Res*. 47, 4845–4860.

Wilkinson, S.J., Shah, N., Pantelides, C.C., 1995. Aggregate Modelling of Multipurpose Plant Operation. *Comput. Chem. Eng*. 19, S583.

Wu, D., Ierapetritou, M.G., 2003. Decomposition Approaches for the Efficient Solution of Short-Term Scheduling Problems. *Comput. Chem. Eng*. 27, 1261.

6

CHAPTER 6

Conclusions and Recommendations

6.1 Conclusions

This work presents different models that address optimization problems for batch plants involving energy, water usage, design, synthesis and medium-term scheduling.

An efficient continuous time mathematical model for direct and indirect heat integration has been presented. Most heat integration models rely on a predefined schedule, which leads to suboptimal results. The contribution presented in this thesis incorporates heat integration into the scheduling framework. Consequently, the model is then capable of solving for both direct and indirect heat integration. By using a heat storage vessel, a considerable reduction in utility consumption is achieved. Application of the proposed formulation with indirect heat integration to an industrial case study showed a 26.6% improvement in objective function compared to standalone operation. Different case studies were used to test the performance of the proposed model and it was found that the model is computationally superior.

It is a common feature for some batch plants to have opportunities for both wastewater minimization through reuse and energy integration. A comprehensive model has been presented where wastewater minimization and heat integration are both embedded within the scheduling framework and solved simultaneously, thereby leading to a truly flexible optimal process schedule. Results from case studies have shown that profit maximization together with heat integration and wastewater minimization give a much better overall economic performance.

Application of the developed method to a particular case has shown that through only applying water integration the cost is reduced by 11.6%, by applying only energy integration the cost is reduced by 29.1% and by applying both energy and water integration the cost is reduced by 34.6%. This indicates that optimizing water and energy integration in the same scheduling framework will reduce the operating cost and environmental impact significantly. A better objective value was achieved using the proposed model when compared to a sequential based method.

A contribution was also made by this work in addressing synthesis and design of batch plants. A recent robust scheduling formulation based on continues time representation is used as a platform for the design problem and achieved a better optimal design and required less computational time. The objective value (profit) obtained using the presented formulation is improved by 228.6% when compared to the recent published formulations for the design and synthesis problem. When this work was compared with other formulations, it results in a smaller size mathematical model that required fewer binary variables, continuous variables and constraints. The model explicitly considered the different location of materials in the plant. The formulation also considered the costs arising from the pipe network and determined the optimal pipe network that should exist between units.

Finally, a medium-term production scheduling technique for multiproduct batch plants has been presented. The developed model is tested with a real case from a pharmaceutical facility. The solution technique for the problem is by the use of a decomposition approach where the large scale problem is divided into tractable size short-term scheduling subproblems. The decomposition model has been implemented to solve a makespan minimization for almost 13 weeks which is difficult to solve when applying the short-term scheduling model for this medium-term time horizon. Additional features of the proposed model are as follows. Storage can be used for different state, a state can be stored in multiple storage units, a state should wait in storage for certain duration before it is used and no batch mixing is allowed in storage. The proposed model can be easily extended to incorporate intermediate due dates for a product.

6.2 Recommendations

i) In the proposed models the due date is assumed at the end of the time horizon. There are also cases where plants are running production on intermediate due dates, incorporation of intermediate due dates in the scheduling model must be investigated.

ii) Further work must be done to address the problem of production scheduling with uncertainty in processing times, market demands, prices of products, cost of raw materials and price of utilities.

iii) Most of the existing literature and the developed models are concentrated on a mono-criterion objective and thus the use of multi-objective models could provide enriched approaches with an increase adequacy to solve real case solutions. Multi stage decisions models where an integration of different decisions at different levels is performed (e.g. energy and water minimization, makespan minimization and optimization of network structure for heat exchanger and piping) should be explored.

iv) The developed model determines the best heat recovery potential. It is assumed that the gain from heat recovery outweigh the cost arises from heat exchanger and heat storage. However, this might not be true for all the cases so the model should further extend to address the best heat exchanger network design as well to generalize in addressing heat integration problem.

v) The design and synthesis model does not account environmental aspects and is still a non-explored area. The development of systematic methods for the design and synthesis problem for multipurpose batch plants that also addressing environmental aspects have an immense value. Almost invariably, this would lead to a very complex problem which would require a scheduling platform with novel features.

vi) The quantification of the uncertain conditions and its impact on the plant design and operation should be further explored. In particular, long-term uncertainties relative to market changes and technology evolution should be studied.

Appendix

The recent robust scheduling model by Seid and Majozi (2012) used in Chapter 2 and Chapter 3 is discussed below. The mathematical model entails sets, parameters, variables and constraints.

Sets

S	$\{S \mid S$ any state $s\}$
I	$\{i \mid i$ is a task$\}$
J	$\{j \mid j$ is a unit$\}$
S_{inJ}^{sc}	$\{s_{inj}^{sc} \mid s_{inj}^{sc}$ task which consume state $s\}$
S_{inJ}^{*}	$\{s_{inj}^{*} \mid s_{inj}^{*}$ tasks performed in unit $j\}$
S_{inJ}	$\{s_{inj} \mid s_{inj}$ is an effective state representing a task$\}$
S_{inJ}^{usc}	$\{s_{inj}^{usc} \mid s_{inj}^{usc}$ task which consume unstable state $s\}$
S_{inJ}^{sp}	$\{s_{inj}^{sp} \mid s_{inj}^{sp}$ task which produce state s other than a product$\}$
S_{inJ}^{usp}	$\{s_{inj}^{usp} \mid s_{inj}^{usp}$ task which produce unstable state $s\}$
$S_{inJ}^{s^p}$	$\{s_{inj}^{s^p} \mid s_{inj}^{s^p}$ task which produce state s which is a product$\}$
S^p	$\{s^p \mid s^p$ a state which is a product$\}$
P	$\{p \mid p$ is a time point$\}$
J_s	$\{j_s \mid j_s$ is a unit producing state $s\}$

Variables

$tp(s_{inj}, p)$	time at which task ends at time point p
$tu(s_{inj}, p)$	time at which task starts at time point p,

$mu(s_{inj}, p)$	amount of material processed by a task at time point *p*
$q_s(s, p)$	amount of state *s* stored at time point *p*
$y(s_{inj}, p)$	binary variable for assignment of task at time point *p*
$t(j, p)$	binary variable associated with usage of state produced by unit *j* at time point *p*
$t(j, s, p)$	binary variable associated with usage of state *s* produced by unit *j* at time point *p* if the unit produces more than one intermediate at time point *p*
$x(s, p)$	binary variable associated with availability of storage for state *s* at time point *p*
$u(s, j, p)$	amount of material stored in unit *j* at time point *p*

Parameters

V_j^U	maximum capacity of unit *j*
$V_{s_{inj}}^U$	maximum capacity of unit *j* to process a particular task
$V_{s_{inj}}^L$	minimum capacity of unit *j* to process a particular task
H	time horizon of interest
QS^o	initial amount of state *s* stored
QS^U	maximum capacity of storage to store a state *s*
$\tau(s_{inj})$	constant term of processing time of task
$\beta(s_{inj})$	coefficient of variable term of processing time of a task
$\rho(s_{inj}^{sp})$	portion of state *s* produced by a task
$\rho(s_{inj}^{sc})$	portion of state *s* consumed by a task

Constraints

Allocation constraints

Constraint (1) implies that at time point *p* only one task is allowed to be performed in unit *j*.

$$\sum_{s_{inj} \in S^{*}_{inj}} y\left(s_{inj}, p\right) \le 1, \ \forall \ j \in J, \quad p \in P \tag{1}$$

Capacity constraints

Constraint (2) implies that the total amount of all the states consumed at time point *p* is limited by the capacity of the unit which consumes the states. $V^{L}_{s_{inj}}$ and $V^{U}_{s_{inj}}$ represent lower and upper bounds in capacity of a given unit that processes the effective state s_{inj}.

$$V^{L}_{s_{inj}} y\left(s_{inj}, p\right) \le mu\left(s_{inj}, p\right) \le V^{U}_{s_{i,j}} y\left(s_{inj}, p\right), \ \forall \ p \in P, \ j \in J, \ s_{inj} \in S_{inJ} \tag{2}$$

Material balance for storage

Constraint (3) states that the amount of material stored at each time point *p* is the amount stored at the previous time point adjusted to some amount resulting from the difference between state *s* produced by tasks at the previous time point (*p*-1) and used by tasks at the current time point *p*. This constraint is used for a state other than a product, since the latter is not consumed, but only produced within the process.

$$q_s(s,p) = q_s(s,p-1) - \sum_{s_{inj} \in S^{sc}_{inJ}} \rho^{sc}_{s_{in,j}} mu\left(s_{in,j}, p\right) + \sum_{s_{inj} \in S^{sp}_{inJ}} \rho^{sp}_{s_{in,j}} mu\left(s_{in,j}, p-1\right), \quad \forall \ p \in P, \quad s \in S \tag{3}$$

Constraint (4) states that the amount of product stored at time point *p* is the amount stored at the previous time point and the amount of product produced at time point *p*.

$$q_s\left(s^p, p\right) = q_s\left(s^p, p-1\right) + \sum_{s_{inj} \in s^{s^p}_{inJ}} \rho^{sp}_{s_{inj}} mu\left(s_{inj}, p\right), \ \forall \ p \in P, \quad s^p \in S^p \tag{4}$$

Duration constraints (batch time as a function of batch size)

Constraint (5) describes the duration constraint modelled as a function of batch size where the processing time is a linear function of the batch size. For zero-wait (ZW) only the equality sign is used.

$$tp(s_{inj}, p) \geq tu(s_{inj}, p) + \tau(s_{inj})\, y(s_{inj}, p) + \beta(s_{inj})\, mu(s_{inj}, p), \quad \forall\ j \in J,\ p \in P,\ s_{inj} \in S_{inJ} \qquad (5)$$

Sequence constraints

The two subsections address the proper allocation of tasks in a given unit that ensures the starting time of a new task to be later than the finishing time of the previous task.

Same task in same unit

Constraint (6) states that a state can only be used in a unit, at any time point, after all the previous tasks are complete. In essence, this implies that a unit must be available before it can be used. It is worth noting that all the tasks referred to in Constraint (6) pertains to the same state, i.e. same task for different batches in the same unit.

$$tu(s_{in,j}, p) \geq tp(s_{in,j}, p-1), \quad \forall\ j \in J,\ p \in P,\ s_{in,j} \in S^{*}_{in,j} \qquad (6)$$

Different tasks in same unit

Constraint (7) states that a task can start in the unit after the completion of all the previous tasks that can be performed in the unit. In the context of Constraint (7), tasks pertain to different states, hence different tasks in the same unit.

$$tu(s_{inj}, p) \geq tp(s'_{inj}, p-1), \quad \forall\ j \in J,\ p \in P,\ s_{inj} \neq s'_{inj},\ s_{inj}, s'_{inj}, \in S^{*}_{inj} \qquad (7)$$

If the state is consumed and produced in the same unit, where the produced state is unstable then in addition to Constraint (7), Constraints (8) is used.

$$tp(s^{usp}_{inj}, p-1) \geq tu(s^{usc}_{inj}, p) - H(1 - y(s^{usp}_{inj} p-1)), \quad \forall\ j \in J,\ p \in P,\ s^{usc}_{inj} \in S^{usc}_{inj},\ s^{usp}_{inj} \in S^{usp}_{inj} \qquad (8)$$

Sequence constraints for different tasks in different

These constraints state that for different tasks that consume and produce the same state, the starting time of the consuming task at time point p must be later than the finishing time of any task at the previous time point p-1 provided that the state is used.

If an intermediate state s is produced from one unit.

Constraints (9) and (10) work together in the following manner:

$$\rho\left(s_{inj}^{sp}\right) mu\left(s_{inj}, p-1\right) \leq q_s(s, p) + V_j^U t(j, p), \forall \ j \in J, \ p \in P, \ s_{inj} \in S_{inJ}^{sp} \tag{9}$$

$$tu\left(s_{inj'}, p\right) \geq tp\left(s_{inj}, p-1\right) - H\left(\left(2 - y\left(s_{inj}, p-1\right) - t(j, p)\right)\right), \forall \ j \in J, \ p \in P, \ s_{inj} \in S_{inJ}^{sp}, \ s_{inj'} \in S_{inJ}^{sc} \tag{10}$$

Constraint (9) states that if the state s is produced from unit j at time point p-1 but is not consumed at time point p by another unit j', i.e. $t(j,p) = 0$, then the amount produced cannot exceed allowed storage, i.e. $q_s(s,p)$. On the other hand, if state s produced from unit j at time point p-1 is used by another unit j' then the amount of state s stored at time point p, i.e.$q_s(s,p)$ is less than the amount of state s produced at time point p-1. The outcome is that the binary variable $t(j,p)$ becomes 1 in order for Constraint (9) to hold. If the unit performs tasks like separation, distillation and other tasks that produce more than one intermediate at time point p then the binary variable $t(j,p)$ becomes $t(j,s,p)$. This allows us at the same time point for Constraint (10) to be relaxed for the unit that is not using the state produced by unit j at time point p. Simultaneously, for the other unit that uses the state produced by unit j at time point p the sequence Constraint (10) holds. Constraint (10) states that the starting time of a task consuming state s at time point p must be later than the finishing time of a task that produces state s at the previous time point p-1, provided that state s is used. Otherwise, the sequence constraint is relaxed.

If an intermediate state is produced from more than one unit

Constraint (11) states that the amount of state s used at time point p can either come from storage, or from other units that produce the same state depending on the binary variable

$t(j,p)$. If the binary variable $t(j,p)$ is 0, which means that state s produced from unit j at time point p-1 is not used at time point p, then Constraint (10) is relaxed. If $t(j,p)$ is 1, state s produced from unit j at time point p-1 is used, as a result Constraint (12) holds. Although Constraint (11) is nonlinear it can be linearized exactly using Glover transformation developed by Glover (1975). It is highly imperative to realize that Constraint (10) plays a pivotal role in both instances when a state is produced from one unit and when a state is produced from many units.

$$\sum_{s_{inj} \in S_{inJ}^{sc}} \rho_{s_{inj}}^{sc} mu\left(s_{inj}, p\right) \leq qs\,(s, p-1) + \sum_{s_{inj} \in S_{inJ}^{sp}} \rho_{s_{inj}}^{sp} mu\left(s_{inj}, p-1\right) t\left(j, p\right) \quad \forall\; j \in J,\; p \in P \tag{11}$$

Constraint (12) states that a consuming task can start after the completion of the previous task.

$$t_u\left(s_{in,j'}, p\right) \geq t_p\left(s_{in,j}, p-2\right) - H(1 - y(s_{in,j}, p-2)),\; \forall\; j \in J,\;\; p \in P,\;\; s_{in,j} \in S_{in,J}^{sp},\;\; s_{in,j'} \in S_{in,j'}^{sc} \tag{12}$$

Sequence constraints for FIS policy

As aforementioned, Type II suboptimlality in previous formulations pertains to modeling for finite intermediate storage (FIS). In most formulations, i.e. Majozi and Zhu (2001), Ierapetritou and Floudas (1998) and Lin and Floudas (2001), this is overlooked and results in unlimited intermediate storage (UIS) behavior in the final schedule. The following constraints are aimed at addressing this drawback. A new binary variable $x(s,p)$ is introduced as shown in Constraints (13) and (14). This binary variable indicates the availability ($x(s,p) = 1$) and absence of storage ($x(s,p) = 0$). According to Constraint (13), any state s can only be stored if the capacity of available storage will not be exceeded. Otherwise, state s will either be produced and consumed immediately or not produced at all. Constraint (14) enforces this condition.

$$\sum_{s_{inj} \in S_{inJ}^{sp}} \rho_{s_{inj}}^{ps} mu\left(s_{inj}, p-1\right) + qs\,(s, p-1) \leq QS^U + \sum_{j \in J_s} V_j^U \left(1 - x(s, p)\right) \;\forall\; j \in J,\; p \in P,\; s \in S \tag{13}$$

According to Constraint (16), the starting time of a task that consumes state s at time point p must be equal to the finishing time of a task that produces state s at time point p-

1, if both consuming and producing tasks are active at time point *p* and time point *p*-1 respectively and if there is no storage to store the amount of state *s* produced at time point *p*. In a case when storage is available to store a state *s* at time point *p*, then the starting time of the consuming task at time point *p* is not necessarily equal to the finishing time of a task producing state *s* at time point *p*-1. Constraints (13) and (14) reduce the number of time points and give a better objective value as can be seen in the case study.

$$tp(s_{inj'}, p-1) \leq tp(s_{inj}, p-1) + H\left(2 - y(s_{inj'}, p) - y(s_{inj}, p-1)\right) + H(x(s,p))$$
$$\forall\ j \in J,\ p \in P,\ s_{inj} \in S_{inJ}^{sp},\ s_{inj'} \in S_{inJ}^{sc} \quad (14)$$

Storage constraints

Constraint (15) indicates that the amount of state *s* stored at any time point must not exceed the maximum capacity of the storage. The state *s* that is produced at time point *p*-1 can be stored for a while in a unit that is producing it in the next time points until it is used, if the unit is not performing tasks.

$$q_s(s,p) \leq QS^U + \sum_{s_{inj} \in S_{inJ}^{sp}} u(s,j,p) \quad \forall\ s \in S,\ p \in P,\ j \in J \quad (15)$$

Constraint (16) states that portion of the state that is produced at time point *p* can be stored in the unit for consecutive time points, if the unit is not active in those consecutive time points. Which is a similar concept to that of slot based formulations that allow task to continue in the next consecutive slots, indicating that the unit stores the states in those consecutive time points.

$$u(s,j,p) \leq \rho_{s_{inj}}^{sp} mu(s_{inj}, p-1) + u(s,j,p-1) \quad \forall\ p \in P,\ \ j \in J \quad (16)$$

Constraint (17) ensures that if a state is stored at time point *p* in the unit then the unit should not be active to start any other task.

$$u(s_{inj}, p) \leq V_j^U \left(1 - \sum_{s_{inj} \in S_{inj}^{*}} y(s_{inj}, p)\right) \quad \forall\ p \in P,\ j \in J \quad (17)$$

Time horizon constraints

The usage and the production of states should be within the time horizon of interest. These conditions are expressed in Constraints (18) and (19).

$$tu(s_{inj}, p) \leq H, \ \forall \ s_{inj} \in S_{inJ}, \ p \in P, \ j \in J \tag{18}$$

$$tp(s_{inj}, p) \leq H, \ \forall \ s_{inj} \in S_{inJ}, p \in P, \ j \in J \tag{19}$$

Tightening constraints

Constraint (20) is used to tighten the model. The sum of duration of all tasks in a unit must be within the time horizon.

$$\sum_{s_{inj} \in S^*_{inj}} \sum_{p} \left(\tau(s_{inj}) y(s_{inj}, p) + \beta(s_{inj}) mu(s_{inj}, p) \right) \leq H, \ \forall \ \ p \in P, \ j \in J \tag{20}$$