WASTEWATER MINIMISATION IN MULTIPURPOSE BATCH PLANTS WITH CHANGEOVER CONSTRAINTS AND ENERGY OPTIMISATION

OMOBOLANLE ADEKOLA
WASTEWATER MINIMISATION IN MULTIPURPOSE
BATCH PLANTS WITH CHANGEOVER
CONSTRAINTS AND ENERGY OPTIMISATION

by

Omobolanle Adekola

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

Ph.D.: Chemical Engineering

in the

Faculty of Engineering, the Built Environment and Information Technology

Supervisor: Prof. Thokozani Majoezi

University of Pretoria

July, 2017
SYNOPSIS

Water and energy are very necessary resources for operating any chemical plant and contribute to total costs. Economic reasons notwithstanding, the processing industry has been incentivised to practise sustainable production in which the consumption of energy and water are more efficient, in response to stringent environmental legislation and public perception.

Published literature exists for the independent investigation of water and energy optimisation in batch plants. It has been established for the individual problems that, a true optimum can only be obtained if batch production scheduling and water use or energy optimisation are performed simultaneously. However, the simultaneous optimisation of both water and energy within the same production scheduling framework has been largely ignored, due to the potential complexities of such a problem. Additionally, literature addressing the minimisation of water in fixed load problems has usually assumed that the water using operations (washing) are sequence independent. This is unlikely, as equipment units usually perform more than one task in multipurpose batch plants. Since the sequence of tasks in a unit influences both the occurrence and extent of washing in the unit, appropriate consideration of task sequences during production can contribute to wastewater minimisation.

This thesis presents a mathematical formulation for the production scheduling of multipurpose batch plants, in which sequence dependent changeover costs are addressed. When compared to an existing formulation, the proposed formulation leads to a smaller sized model with fewer binary variables, continuous variables and constraints for a given case study, although the same objective was obtained. Expanding upon this, a mathematical formulation for simultaneous batch production scheduling and wastewater minimisation, for which the water requirement in a unit is dependent on tasks and their successors, is presented. The effectiveness of the formulation was demonstrated using two case studies. The results show improvements in profit and reduced wastewater generation when the sequence of tasks is taken into consideration. One case study saw water savings of 48% achieved with this method. The formulation was extended to incorporate process integration in the form of direct water reuse, which resulted in a further improvement in profit and water use.
The third contribution in this thesis is a simultaneous method for the optimisation of energy and water embedded within a scheduling framework. In addition, opportunities for direct and indirect heat integration as well as direct and indirect water reuse were explored with the objective of improving the profitability of the plant while minimising water and external utility usage. The applicability of the method was demonstrated with three case studies. The developed formulation proved superior to a method that solved the same problem sequentially.

*Keywords*: batch scheduling, sequence dependent changeover, wastewater minimisation, energy optimisation
LIST OF PUBLICATIONS

CONFERENCE PUBLICATIONS


JOURNAL PUBLICATIONS


DECLARATION

I, Omobolanle Adekola, student number 04320050, declare that:

1. I understand what plagiarism is and am aware of the University’s policy in this regard.
2. I declare that this thesis is my own original work. Where other people’s work has been used (either from a printed source, the internet or any other source), this has been properly acknowledged and referenced in accordance with departmental requirements.
3. I have not used work previously produced by another student or any other person to hand in as my own.
4. 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.
5. The work presented in this dissertation has not been submitted anywhere else in partial or full fulfillment of another degree.

Student___________________
ACKNOWLEDGEMENT

I would like to thank God for the ability and wherewithal to accomplish this task.

My deepest thanks to my supervisor, Prof. T. Majozi for your patience, guidance, corrections and the opportunities that you provided for me to improve myself. It is a privilege to have been one of your students.

I express my heartfelt thanks to my husband, Prof. T. Bello-Ochende for your constant support and my sons Adinoyi and Adavize, born during my research programme for teaching me patience and the value of flexibility. My sincere gratitude goes to the Olorunju family, for your constant well wishes and words of encouragement. Finally, to my family and friends, thank you for your unconditional love.
CONTENTS

CHAPTER 1 Introduction ........................................................................................................ 11
  1.1. Background ............................................................................................................. 11
    1.1.1. Significance of changeover for wastewater minimisation .............................. 11
    1.1.2. Minimisation of energy and water through process integration .................... 12
  1.2. Motivation for the research...................................................................................... 12
  1.3. Research objectives ............................................................................................... 13
  1.4. Thesis structure ..................................................................................................... 13
  References ....................................................................................................................... 14

CHAPTER 2 Literature Review .............................................................................................. 16
  Introduction ...................................................................................................................... 16
  2.1. Batch processing .................................................................................................... 16
    2.1.1. Multiproduct vs. multipurpose batch plants .................................................... 16
    2.1.2. Intermediate storage policy ............................................................................. 17
  2.2. Batch scheduling .................................................................................................... 18
    2.2.1. Overview of scheduling methodologies ............................................................ 19
    2.2.2. Features of MILP scheduling formulations ....................................................... 20
    2.2.3. MILP batch scheduling formulations ............................................................... 25
    2.2.4. Scheduling formulations addressing changeover ............................................... 32
  2.3. Process integration ................................................................................................. 37
    2.3.1. What is ‘Pinch Analysis’? ............................................................................... 37
    2.3.2. Application of pinch analysis to other areas ...................................................... 38
    2.3.3. Process integration techniques for energy and water in literature .................... 40
    2.3.4. Heat integration in batch plants ...................................................................... 41
    2.3.4. Wastewater minimisation in batch plants ....................................................... 50
    2.3.5. Optimisation of both energy and water in batch plants .................................... 59
  2.4. Research gaps ......................................................................................................... 60
  References ....................................................................................................................... 61

CHAPTER 3 A New Scheduling Formulation for Multipurpose Batch Plants with Sequence Dependent Changeover ............................................................................................... 78
  Introduction ...................................................................................................................... 78
  3.1. Problem statement .................................................................................................. 78
5.2.2. Wastewater minimisation constraints .......................................................... 138
5.3. Solution procedure .......................................................................................... 148
5.4. Case study I (Halim and Sinivasan, 2011) ....................................................... 151
5.5. Case study II .................................................................................................... 154
5.6. Case study III .................................................................................................. 159
5.7. Conclusions ..................................................................................................... 164
Nomenclature ........................................................................................................... 165
References .................................................................................................................. 169
CHAPTER 6 Conclusions and Recommendations ..................................................... 171
6.1. Conclusions ..................................................................................................... 171
6.2. Recommendations for future work ................................................................... 173
LIST OF FIGURES

Figure 2.1. Different time representations ................................................................. 22
Figure 2.2. Composite curves showing heat integration .............................................. 38
Figure 2.3. Limiting composite curve (Wang and Smith, 1994) ................................. 39
Figure 3.1. STN (a) and SSN (b) representations of case study ............................... 93
Figure 3.2. Resulting production schedule for case study ......................................... 95
Figure 4.1. Motivation for the study: (a) Sequence independent washing (b) Sequence dependent washing ................................................................. 102
Figure 4.2. Superstructure for mathematical formulation ......................................... 104
Figure 4.3. Superstructure including direct reuse ...................................................... 108
Figure 4.4. STN (a) and SSN (b) representation of case study I ............................. 113
Figure 4.5. Scenario 1, case study I ......................................................................... 115
Figure 4.6. Scenario 2, case study I ......................................................................... 116
Figure 4.7. Scenario 3, case study I ......................................................................... 116
Figure 4.8. STN (a) and SSN (b) representation of case study II .......................... 117
Figure 4.9. Scenario 1, case study II ....................................................................... 119
Figure 4.10. Scenario 2, case study II ...................................................................... 120
Figure 4.11. Scenario 3, case study II ...................................................................... 120
Figure 4.12. Scenario 1, case study III ................................................................. 122
Figure 4.13. Scenario 2, case study III ................................................................. 122
Figure 4.14. Scenario 4, case study III ................................................................. 123
Figure 4.15. Scenario 5, case study III ................................................................. 123
Figure 5.1. Superstructure for mathematical formulation: (a) when units perform heating/cooling tasks, (b) when units perform washing tasks. ................. 130
Figure 5.2. Recipe representation of the simple sequential process for case study I .... 151
Figure 5.3. Resulting production schedule for case study I .................................... 154
Figure 5.4. STN representation of the complex production facility for case study II .............................. 155
Figure 5.5. Resulting production schedule for case study II .................................. 157
Figure 5.6. Resulting process schedule when only direct heat integration and direct water reuse are possible ................................................................. 163
Figure 5.7. Resulting process schedule when both direct/indirect heat integration and wastewater minimisation are possible ........................................ 164
LIST OF TABLES

Table 3.1. Production data ....................................................................................................... 94
Table 3.2. Utility requirements ............................................................................................... 94
Table 3.3. Cleaning data .......................................................................................................... 94
Table 3.4. Computational statistics of case study .................................................................... 96
Table 4.1. Production data for case study I ............................................................................ 114
Table 4.2. Changeover times for case study I ......................................................................... 114
Table 4.3. Water using data for case study I .......................................................................... 114
Table 4.4. Computational statistics of case study I ............................................................... 116
Table 4.5. Data pertaining to production for case study II (Halim and Srinivasan 2012) ......... 118
Table 4.6. Changeover times for case study II ....................................................................... 118
Table 4.7. Data pertaining to water requirements for case study II ....................................... 118
Table 4.8. Computational statistics of case study II .............................................................. 121
Table 4.9. Water using data for case study III ...................................................................... 121
Table 4.10. Computational statistics of case study III ........................................................... 124
Table 5.1. Data pertaining to production, for case study I ..................................................... 151
Table 5.2. Data pertaining to energy requirements, for case study I ..................................... 151
Table 5.3. Data pertaining to water requirements, for case study I ....................................... 152
Table 5.4. Comparison of results for case study I ................................................................. 153
Table 5.5. Data pertaining to production, for case study II ................................................... 156
Table 5.6. Data pertaining to energy requirements, for case study II ................................... 156
Table 5.7. Data pertaining to water requirements, for case study II ....................................... 157
Table 5.8. Comparison of results for case study II ............................................................... 158
Table 5.9. Production data for case study III ......................................................................... 159
Table 5.10. Wastewater minimisation data for case study III ............................................... 160
Table 5.11. Heat integration data for case study III ............................................................... 160
Table 5.12. Comparison between different scenarios for case study III............................... 162
CHAPTER 1

Introduction

1.1. Background

Batch processes are sought after in manufacturing environments for their characteristic flexibility in the form of shared equipment and resources for production. These industries are faced with the pressure to increase product offerings, manage the variability in demand for products and improve customer response time, while being competitive in price. Examples are the pharmaceutical, agrochemical, paint, dairy, and food industries. Capitalising on the advantageous features of batch processes presents some challenges due to the fact that they are by nature discrete with finite durations.

Given that time is such an important factor affecting production, batch scheduling is crucial. According to Floudas and Lin (2004), scheduling in batch plants aims to determine when, where and how to produce a set of products given a limited set of resources, the processing recipes and the specific time horizon. This subject has received a lot of attention from both industry and academia over the past few decades. Existing methods can differ in the types of batch plants considered, the objectives addressed, the way time is handled, and so on. Studies of these different characteristics can be found in Burkard and Hatzl (2005), Floudas and Lin (2004), Li et al. (2010), Mendez et al. (2006), Pan et al. (2009) and Shaik et al. (2006).

1.1.1. Significance of changeover for wastewater minimisation

Due to the nature of multipurpose batch plants, equipment units usually perform more than one task. Changeover involves any necessary preparation required for a succeeding task to take place in a unit after the completion of the current task. Changeover can be sequence dependent or independent. When changeover is sequence dependent, changeover time, changeover cost or both, depend on the task just completed as well as the task about to be conducted in a unit. Frequent changeovers result in reduced equipment availability and increased consumption of necessary resources namely, water for washing, both of which can negatively affect plant productivity.
1.1.2. Minimisation of energy and water through process integration

Energy and water are ubiquitous in operating chemical plants. Operations which convert raw materials to final product can consume large amounts of energy, due to the requirement to satisfy proper temperatures for reactions to occur. Likewise, water serves many uses such as, a raw material for production, a medium for the transport of energy, an agent for cleaning, etc. The use of natural gas and fuel oil as sources of energy is associated with greenhouse gas emissions (Fernandez et al., 2012) and although water is relatively cheap, batch processes generate effluents with toxic contaminants that can be expensive to treat (Gouws et al., 2010). For these reasons and others, process integration techniques were developed to reduce environmental pollution associated with the consumption of energy and water during plant operation (Liu et al., 2013).

1.2. Motivation for the research

Equipment cleaning with water for the changeover operation is of particular interest. The order in which tasks take place in a unit affects the amount of water required for washing as exemplified by the following scenario associated with the paint industry. Suppose a batch of white dye and a batch of black dye are to be processed in a unit suitable to process both dyes. If the black dye is processed first, water must be used to wash the unit thoroughly at the end of the process to prepare the unit for the white dye. A large amount of water will be required and consequently, a large amount of wastewater will be generated. In contrast, if the white dye is processed first in the unit, the amount of cleaning water necessary to prepare it for processing the black dye will be less. Given that the order in which operations take place in a unit can contribute to the amount of washing water required, the task sequence in a unit should not be disregarded.

Water and energy optimisation in batch plants have been treated as separate problems in literature. Halim and Srinivasan (2011) developed one of the earliest methods to address batch production together with the optimisation of water and energy, by taking advantage of the fact that alternative batch schedules exist for a given optimal production objective. In that work, the batch production schedule was optimised first and alternate schedules corresponding to the optimal solution were generated through a stochastic search-based
integer cut procedure. Next, these alternate schedules were used to optimise the energy and water consumption with the aim of retaining the optimality of the scheduling solution. A more unified approach is desirable to optimise both water and energy usage within the batch production scheduling problem by ensuring that results that could have been overlooked when time is fixed are available during the optimisation search, leading to better overall objectives.

1.3. Research objectives

The above discussion highlights the research interests of this thesis, the objectives of which are:

- to develop a scheduling formulation for multipurpose batch plants that effectively addresses sequence dependent changeover costs,
- to extend the scheduling formulation just described, for the simultaneous optimisation of batch production and water use in a multipurpose batch plant, in which the water requirement is determined by the sequence of tasks in units and
- to develop a mathematical formulation for the simultaneous optimisation of energy and water use within a unified multipurpose batch scheduling framework.

1.4. Thesis structure

This thesis is divided into six chapters. The next chapter, Chapter 2 is devoted to a literature review on the following subjects: scheduling in batch plants with emphasis on sequence dependent changeover and the optimisation of energy and water in batch plants. Chapter 3 presents a mathematical formulation for batch production scheduling investigating a number of notable features, chief of which are sequence dependent changeover costs. Chapter 4 presents a mathematical formulation for the simultaneous optimisation of batch production and water use in a multipurpose batch plant, taking into consideration the effects of sequence dependent changeover on the water usage. Chapter 5 presents a unified mathematical formulation for the simultaneous optimisation of batch production and minimisation of energy usage and wastewater generation. Finally, Chapter 6 discusses the conclusions and recommendations drawn from the research.
References


CHAPTER 2

Literature Review

Introduction

This chapter discusses the research efforts that have contributed to the subject matters addressed in this thesis. The literature review is divided into three broad sections. A brief introduction to batch processing is discussed first to provide the foundation necessary to appreciate the different concepts and approaches encountered in literature. Next, scheduling of batch plants, a subject of significance is explored, concluding with changeover considerations. Finally, process integration as it pertains to water and energy optimisation in batch plants is investigated.

2.1. Batch processing

Continuous processes are designed to operate daily throughout the year with the exception of allocated downtime for the repeated production of bulk volumes of identical products, e.g., petrochemicals. Consequently, input and output materials are always present. On the other hand, batch processes are designed to operate intermittently with distinct starting and finishing times, for the manufacture of low-volume, high-value added products. As a result, material flows do not always exist. Batch plants can employ a variety of production regimes during their operation which influence how much material is to be processed and in what unit, what processing sequence must be adopted in a unit, etc. This allows batch plants to be adaptable to fluctuating product demands and more flexible than their continuous counterparts.

2.1.1. Multiproduct vs. multipurpose batch plants

Batch plants can be broadly classified as multiproduct or multipurpose based on the flow of product throughout the plant. In multiproduct plants also known as “flow-shop”, all product
batches follow the same equipment path. Usually, one product is produced at a time in batches and different products are manufactured using the same equipment in successive production runs or campaigns. In multipurpose plants also known as “job-shop”, multiple batches of the same product can use different pieces of equipment and do not need to follow the same equipment path. Also, a multipurpose batch plant allows for the simultaneous production of multiple batches of different products that have different recipes. Multipurpose plants are the more complex of the two.

2.1.2. Intermediate storage policy

Another classification of batch plants is based on the operational and transfer policies between equipment with respect to intermediate material storage. Storage introduces flexibility to batch plant operations to mitigate capacity and time constraints. The transfer policies are described below:

- **Fixed intermediate storage, FIS**, takes advantage of the storage of intermediate material in a vessel of finite capacity before it is used in successive tasks. However, due to the limited capacity of finite storage, storage capacity is not guaranteed.
- **Unlimited intermediate storage, UIS**, implies that storage capacity for material is always available.
- **Common intermediate storage, CIS**, implies that storage vessels are shared by more than one task in the batch plant, as opposed to a situation where storage vessels are dedicated to unique tasks.
- **No intermediate storage, NIS** applies to circumstances whereby no storage vessels exist for intermediate material.
- **Process intermediate storage, PIS** refers to the storage of intermediate material in a processing unit that is idle.
- **Zero wait, ZW** applies in situations whereby the intermediate material is to be used immediately after it is produced. An example is an unstable intermediate material, storage of which is undesirable.
- **Finite wait, FW**, applies in other instances where the intermediate can be stored either in the unit producing it or a storage vessel for a given period of time, exceeding which, it becomes unstable.
• Unlimited wait, UW, refers to the situation where intermediate material can be stored for a significant length of time either in the unit producing it or a storage vessel.
• Mixed intermediate storage, MIS, applies in situations where more than one type of the aforementioned intermediate storage philosophies is encountered. For example, in the same batch plant, some tasks have dedicated storage of finite capacities, while other tasks share a common storage.

2.2. Batch scheduling

According to Rippin (1993), a batch processing system for manufacturing products has three necessary components: a production recipe which describes the various operations or tasks required to convert raw material into product, a plant topology which consists of the set of equipment suitable for the tasks and a market which determines the timings and amounts of product required.

The flexibility attributed to batch plants makes the management of batch processes a challenging endeavor. From the variety of production regimes possible, a choice must be made for the procedure that allows the batch plant to operate at the highest possible productivity level. Batch scheduling addresses this problem.

Batch scheduling as formally defined by Pekny and Reklaitis (1998) is a procedure which determines the optimal sequence of different tasks on the available units, the batch size of each task performed on each unit, the times at which these tasks begin and end, given the suitability of the various units (processing and storage) for tasks, recipe for production and a specific objective. Honkomp et al. (2000) observed that in the past, scheduling a batch plant involved the use of manual, error-prone spreadsheet tools which were executed in an ad-hoc approach and were limited by human experience. The possibility that optimality was not achieved was ever present, resulting in underutilisation of available resources and inadequate productivity. This opportunity to improve plant productivity and profitability facilitated the development of advanced systematic methods, which were often computer-aided (Roe et al. 2005). The considerable variety of plant capabilities and requirements has contributed to the extensive research in batch scheduling in the recent decades.
2.2.1. Overview of scheduling methodologies

The scheduling problem in batch processes involves a large number of discrete decisions and alternative solutions which renders scheduling as combinatorial in nature and challenging from a computational perspective. The scheduling problem falls under the category of NP-complete problems (Applequist et al., 1997). As a result, execution times for such problems are exponential to the problem size. Due to the computational time associated with the batch scheduling problem, heuristic procedures were proposed to reduce solution times as Kelly and Mann (2004) noted. In these methods, assumptions were applied to reduce the size of the solution space but, solution optimality could not be guaranteed.

The use of mathematical programming to solve exact algorithms of scheduling problems was adopted due to the advantage that optimum solutions could be found. The drawback of these methods was long solution times. Recently, due to advancement in computational technology, mathematical programming methods have become popular. These techniques are linear programming, LP, integer programming, IP, mixed integer linear programming, MILP, mixed integer non-linear programming, MINLP, constraint programming, CP and hybrids e.g. of CP and MILP, hybrids of heuristic and exact algorithms, etc. The structure/type of problem usually dictated the type of mathematical programming method adopted. However, for complex problems of large sizes, some mathematical programming formulations were unable to determine an optimal solution in reasonable computational time. This led to the development of local search based methods, which are generally less computationally expensive than their mathematical programming counterparts. These methods are genetic algorithm, GA, simulated annealing, SA, tabu search, TS, etc. Nevertheless, these methods often gave suboptimal results.

The batch scheduling problem has been solved most commonly using mixed integer linear programming. This is because an MILP has a guaranteed global optimum solution. For this reason, the remainder of the review on batch scheduling will be exclusive to MILP formulations. The important features required to formulate a scheduling problem as a MILP are discussed below.
2.2.2. Features of MILP scheduling formulations

2.2.2.1. Time representation

Mathematical scheduling formulations are characterised by their representation of time. This is because, in batch plants, resource utilisation is discontinuous by nature. Hence, it is necessary to keep track of the instances along the time horizon where a task begins/ends or a resource is utilised. The time representation can be classified as discrete or continuous.

Discrete time representation

In the discrete time models, the time horizon is divided into a fixed number of intervals of known and equal length, with a binary variable associated with each interval. Events, such as the starting or finishing time of tasks can only take place at the boundaries of these intervals. Therefore, in order to represent a process accurately, the length of each interval is usually determined by the smallest duration of all processing tasks. This leads to large problem sizes for industrial applications. To circumvent this, rounding of duration is used and the number of binary variables decreases. However, the model is not accurate and the results from programming are suboptimal or infeasible. On the other hand, an advantage of the discrete representation is that it provides a common time grid for all tasks, making material balances involving shared resources straightforward to represent. This common time grid is also referred to as having global time points. Early scheduling attempts such as Kondili et al. (1993) and Shah et al. (1993) relied on the discrete representation of time and typically addressed problems with fixed processing times.

Continuous or non-uniform time representation

Continuous time formulations were developed to overcome the limitations of discrete formulations. The earliest of such publications was by Zhang and Sargent (1996). In continuous time models, events can occur at any point during the time horizon, thus eliminating unnecessary time intervals and leads to smaller problem sizes. Due to the fact that these time/event points are not fixed, constraints which match time/event points with the start or finish of a task are necessary. These additional constraints are usually in the form of big-M constraints and cause the mathematical model to have complicated structures leading to
increased integrality gaps unlike models based on discrete time as Floudas and Lin (2004) observed. The continuous time representation can accommodate instances where processing times are not fixed. The continuous time representation can be broadly classified as event based or slot based continuous time representations. As explained by Floudas and Lin (2004), event points are a sequence of time instances located along the time axis and signify the start of a task or the utilisation of a unit. On the other hand, for slot based representations, the time horizon is divided into consecutive blocks of slots of unequal and unknown lengths. $N$ slots is equivalent to $n + 1$ events. A batch task is assigned to a set of consecutive time slots. Examples are the works of Pinto and Grossmann (1995) and Karimi and McDonald (1997).

The number of event points/time slots required to find optimal solutions is a very important performance indicator for slot based and event based continuous time formulations. The continuous time representations can be classified even further with respect to the relationship between time points and tasks in units. The differences between these classifications are illustrated in Figure 2.1 and are explained as follows:

- **Global events or synchronous time slots or single time grid:**
  Here the time grid (set of time points/events) is common to all units. If event points (or slots) are defined at the beginning of each task in each unit, the beginning of a task in a unit triggers an event point (the start of a slot) across all units. This means that for industrial scale problems or those involving many small task durations, a large number of event points is required to solve the problems to optimality. This in return leads to large CPU times. However, when an intermediate material or common resource is shared between two units, global events (synchronized slots) are preferred due to the straightforward manner to handle material balances and resource usage. Examples of models with global event points are Castro et al. (2004), Mockus and Reklaitis (1999) and Zhang and Sargent (1996). Examples of synchronous slot based models are Schilling and Pantelides (1996) and Sundaramoorthy and Karimi (2005).

- **Unit specific events or asynchronous time slots or multiple time grid:**
  Here, the number of time grids is equivalent to the number of units. This format allows tasks which correspond to the same event point but taking place in different units to occur at different times on the time horizon. The relative timings of tasks on various units and the relationship between the mass balances in these units need to be
determined through the use of special sequencing constraints. Examples of unit specific event based formulations are Shaik, Janak and Floudas (2006) and Seid and Majozi (2012). Examples of asynchronous slot based models are McDonald and Karimi (1997) and Lim and Karimi (2003).

Figure 2.1. Different time representations

Time based formulations have the following disadvantage: Before solving a model based on the time grid, the number of event points (slots) must be specified by the user. As such, the appropriate number required for optimal solution is unknown beforehand. Too many event points can make the problem intractable while using too few, will lead to suboptimal results. The number of event points required to solve a problem to optimality is usually obtained through an iterative process. A small number of event points is assumed first and the solution
of the problem is recorded. The event points are increased by one and again the solution is noted. This procedure is repeated until no further improvement in the objective is observed. The smallest number of event points that yields the best objective function is the optimum number of event points that is reported.

2.2.2.2. Recipe representation

The mathematical scheduling formulation captures the characteristics of the production process investigated, in terms of its recipe and material balances. The recipe representation can be broadly classified into network based and sequential based representations. The recipe representation dictates the type of constraints that are formulated.

Network based representation

This is the more general of the two categories and is capable of encapsulating the complex features of many batch plants including production recipes involving multiple products with dissimilar production paths and the splitting or merging of batches. The explicit description of material balances is required. The network based representation usually leads to larger sized models as a result of their generality and are often formulated for problems with shorter time horizons and fewer processing tasks. The network based representations this author has encountered in literature are the following:

- State task network representation, STN:
  Kondili et al. (1993) introduced the STN representation of product recipes which illustrates the transformation of raw materials into final products with the use of states, represented by circles, tasks represented by rectangular blocks and flow of material represented by directed arcs. A state is any material (raw material, intermediate, final product and waste) with distinct properties and attributes.

- Resource task network representation, RTN:
  The RTN was introduced soon after by Pantelides (1994) to include resources. Here, circles represent states (as defined in the STN) and resources such as processing units, storage vessels and utilities. The resources are described as being consumed at the beginning of a task and produced at the completion of the task. In addition to explicit
mass balance equations, formulations based on the RTN include explicit resource balance equations. The RTN is especially advantageous for problems involving identical equipment. Mendez et al. (2006) explain that for these types of problems, the RTN model introduces a single binary variable as opposed to the multiple variables required by the STN model.

- State sequence representation, SSN:
  Majozi and Zhu (2001) developed the SSN representation, a condensed form of the STN. Only states are represented, thereby eliminating the need for task and unit binary variables as required by the STN representation. A task is implied between two different states. The states (as defined in the STN) are represented by circles. The arc linking one state to another implies that a task has taken place and represents the flow of material

*Sequential based representation*

The formulations in this category take advantage of the fact that some batch plants consist of sequential processes, whereby; different products follow the same processing path in a single stage or multiple stages. Such problems are batch or order oriented and do not require the definition of states or tasks. Hence, the explicit consideration of material balances is not required. However, it is necessary to pre-postulate the possible number of batches to be produced. It is interesting to note that these formulations do not rely on concepts such as time slots or event points. Although continuous variables to capture the timing of batches are defined, these times are not assigned to a grid. Sequencing binary variables are defined to relate the precedence of two batches on a unit and address the timing of batches. Since time slots are not needed, these models need to be solved only once to give a global solution if solved to zero optimality. This is an advantage sequential models have over continuous time models. Sequential based formulations are computationally less expensive than network based formulations, although less general. These formulations can be further classified below.

- Global or general or indirect precedence:
  The global sequencing binary variables are active for all pairs of orders regardless of whether they are assigned to the same unit or whether they are consecutive. One sequencing binary variable for each pair of batch tasks that can be allocated to the
same unit is defined. The model leads to fewer binary variables but is incapable of addressing sequence dependent changeover or setup costs. The formulations of Mendez et al. (2001) and Mendez and Cerda (2003) are examples.

- Immediate or direct precedence:
  Describes the relationship between each pair of consecutive orders without taking into consideration whether these orders are assigned to the same unit. Two sets of binary variables are required to define allocation and sequencing decisions. Gupta and Karimi (2003) is an example of such a formulation.

- Unit specific immediate precedence:
  Extends the concept of immediate precedence by considering consecutive orders assigned to the same unit. A single binary variable defines the allocation and sequencing of a pair of orders. Cerda et al. (1997) is an example.

- Unit specific global precedence:
  Introduced by Kopanos et al. (2009), combines the concepts of global precedence and unit specific immediate precedence in order to effectively handle sequence dependent changeover.

2.2.3. MILP batch scheduling formulations

As aforementioned, a substantial number of MILP scheduling formulations have been developed over the past three decades. A major concern has been to improve generality with respect to the types of problems a formulation can solve. The number of constraints and binary variables in a mathematical model has a significant impact on the computational performance of a given model. As Li et al. (2010) observed, in general, the model solution times grow exponentially with the number of binary variables. Consequently, another point of interest was the efficiency of models with respect to decreased solution times, and better computational results in some cases.

The most important contributions made to the scheduling problem in literature have been well cited and adopted for the investigation of other concerns affecting batch plants or adapted by authors for further improvement. These contributions are discussed forthwith.
2.2.3.1. STN scheduling formulations

Kondili et al. (1993) presented their seminal work introducing the STN representation of production recipes for scheduling general multipurpose batch plants. That formulation was based on the discrete time representation. A binary variable $w_{ijt}$ was used to allocate tasks to units. It had a value of 1 if a task $i$ started in a unit $j$ at time $t$. The number of binary variables associated with this representation was $i \times j \times t$ and often led to an explosive binary dimension for complex case studies and were computationally expensive to solve.

Pinto and Grossmann (1995) proposed a continuous slot based formulation to circumvent the difficulties of the discrete time formulations. The formulation was derived for plants with multiple stages in parallel. To assign orders to slots, a tetra index binary variable was defined. This led to a large number of binary variables and an associated computational difficulty for the models. For large case studies, a heuristic rule for the pre-ordering of batches in units was employed.

Ierapetritou and Floudas (1998) presented a landmark scheduling formulation in which the task events were decoupled from the unit events of the STN. Two binary variables, $wv(i,t)$ and $v(j,t)$ were used to define the beginning of a task and unit event respectively. The number of binary variables in this formulation was $(i \times j) + t$ which was less than $i \times j \times t$. This led to a smaller model. Their formulation also exploited the one to one relationship between units and tasks (a unit was allocated to a specific task) in a bid to decompose the problem to a smaller size. This decomposition was not possible in the absence of the one to one relationship. Another important contribution of this work was the introduction of unit specific event points. No other resources apart from materials and equipment were considered. The model required fewer time points compared to competitive global slot based and event based formulations of the time, Schilling and Pantelides (1996) and Zhang (1995) respectively.

The unit specific continuous time formulation of Giannelos and Georgiadis (2002) was similar to that of Ierapetritou and Floudas (1998) except for a few important differences. Special sequencing restrictions were placed on the starting and finishing times of tasks consuming and producing the same intermediate product, forcing them to be the same. Task
durations were relaxed with the use of buffer times, thereby, eliminating the big-$M$ constraints of Ierapetritou and Floudas (1998). This feature was for the applicability of their formulation to continuous processes. The formulation has been shown by Shaik et al. (2006) and Sundaramoorthy and Karimi (2005) to obtain suboptimal results.

Maravelias and Grossmann (2003) developed a global continuous time based representation that accounted for resources such as utilities. In addition, various storage policies like FW, ZW, UIS were covered. Tasks were allowed to continue over multiple time points in this work. For this reason, three binary variables were employed, $w_{r_i,n}$ to signify the start of a task $i$ at event point $n$, $w_{f_i,n}$ to signify the end of a task $i$ at event point $n$ and $w_{p_i,n}$, to indicate that a task was being processed at event point $n$. It has been observed by many researchers to date that this formulation leads to large problems with long solution times.

Janak et al. (2004) extended the formulation of Ierapetritou and Floudas (1998) to address features such as shared storage policy, sequence dependent changeover and additional resource constraints for limited utility availability. Similar to Maravelias and Grossmann, tasks were allowed to continue over several consecutive time points in order to accurately account for the utilisation of the different storage policies and resources. Storage of material was achieved through the use of storage vessels and idle units in the form of storage tasks. The formulation was also adapted to sequential batch problems with the introduction of order satisfaction constraints. As such, objective functions such as the minimisation of order earliness could be accommodated. A drawback of their formulation was that it resulted in large models with excessive computational times due to large numbers of binary and continuous variables.

Sundaramoorthy and Karimi (2005) developed a synchronous slot based formulation that did not include the decoupling of tasks from units. Synchronous slots were employed for the simplified treatment of shared resources, even though asynchronous slots generally lead to fewer slots. A novel concept of time balances, uncommon for slot based formulations was introduced. Binary variables which denoted the start of a task in a unit at a slot were defined. In addition, tasks could finish before the end of a slot and were allowed to span over multiple slots. For this, a continuous 0-1 variable, $y_{ijk}$, was defined to describe whether a unit $j$ continued to perform a task $i$ at time $T_k$. As a result, computation of the remaining processing
time required to complete a task that was in progress during a slot on a unit, was possible. An additional zero task was defined to model the idling of units and also to occupy redundant slots. The formulation proved to be more efficient especially in solving makespan minimisation problems, than models in the same class, such as, Giannelos and Georgiadis (2002) and Maravelias and Grossmann (2003) both of which employed the decoupling of tasks from units.

Shaik et al. (2006) presented a comprehensive comparison of the performance of some unique and proficient continuous-time formulations for problems involving UIS. These were the unit specific event based STN of Giannelos and Georgiadis (2002), global time slot based RTN of Castro et al. (2004), global event based STN of Maravelias and Grossmann (2003), global slot based STN of Sundaramoorthy and Karimi (2005) and a modification to the formulation of Ierapetritou and Floudas (1998) in the form of elimination of dependent variables and combination of constraints. The model statistics used for the comparison were problem size (number of binary and continuous variables), computational times and nodes required to reach zero integrality gap, all performed on the same computer. The objective functions for these comparisons were makespan minimisation and profit maximisation. The general observations made were that both the global event based and slot based models always required the same number of event points (since $n$ slots is equivalent to $n + 1$ events), while the unit specific event model of Shaik et al. (2006) required less event points to solve a problem to global optimality. They also found that out of all the global slot or event based models compared, Castro et al. (2004) performed the best. They concluded that the unit specific event representation was responsible for the superiority of their model.

The main purpose of the formulation by Shaik and Floudas (2009) was to provide a unified, generic framework capable of solving complex scheduling problems with additional resource constraints (for utilities, manpower, etc.). The authors improved on the formulation of Janak et al. (2004), which was developed to consider resource constraints but required significant computational time. Shaik and Floudas (2009) emphasized the need for tasks to be allowed to span multiple event points. For this, a user defined parameter, $\Delta n$ such that $n \leq \bar{n} \leq n + \Delta n$, was introduced to control the number of event points that a task could span. Iteration was performed on this parameter for every specified number of event points to obtain an optimal solution. The model utilised a 3-index binary variable, $w(i, n, \bar{n})$ to indicate whether a task $i$
which started at event point $n$, ended at $\hat{n}$. A drawback of the formulation is the computational time and effort required to iterate over $\Delta n$ to get optimal solutions. However, the authors compared it to the sum of computational times involved during the iteration required to determine the number of time points or slots that yields the optimum solution used in the formulations of Castro et al. (2004) and Sundaramoorthy and Karimi (2005). This study also revealed a few interesting points. For some problems involving profit maximisation or makespan minimisation, Castro et al. (2004) performed better in terms of computational time. This led to the conclusion that certain formulations perform better than others when considering certain classes of problems.

2.2.3.2. RTN scheduling formulations

Pantelides (1994) introduced the RTN by extending the concept of states in the STN to include resources. Like Kondili et al. (1993), the formulation also relied on the discrete time, with the associated disadvantage of excessively large models for finer time discretisations.

Zhang and Sargent (1996) employed a global continuous time representation for the purpose of reducing the number of binary variables. The formulation was an MINLP formulation that was linearised exactly using Glover transformation (Glover, 1975). This linearisation introduced additional variables and constraints, adding to the size of the model.

Castro et al. (2001) presented a scheduling formulation also based on global continuous-time slot representation. Due to the nature of the time representation, no special sequencing constraints were involved which led to the absence of big-$M$ constraints except for those relating extent of tasks to the corresponding binary variables. In a subsequent publication, Castro et al. (2004) presented an improved model by eliminating redundant binary and continuous variables and introducing new timing constraints. This resulted in fewer constraints and better LP relaxed solutions compared to their previous model. Furthermore, in this new formulation, tasks could span multiple time slots. Hence, a new parameter, $\Delta t$, was introduced, which defined a limit on the number of slots over which a task could span. The choice of $\Delta t$ had a profound impact on the solutions obtained, requiring iterations over the parameter to be performed in order to obtain global optimal solutions. An advantage of the formulations by Castro and co-workers (2001, 2004) over unit specific event based or
multiple time grid ones was that due to the nature of the global time grid used, problems with FIS did not lead to infeasible solutions due to the fact that the real time associated with event points was consistent for all units. In unit specific event or multiple time grid models, storage violations in real time were observed due to the fact that the same event point on units transferring (or receiving) material from storage did not correspond to the same real time. As a result, although constraints associated with material storage were feasible where event points were concerned, the equations became infeasible where real time was concerned.

The STN models of Ierapetritou and Floudas (1998) and Shaik et al. (2006) were applicable to scheduling formulations with UIS and problems with no additional resource constraints. Shaik and Floudas (2008) proposed an RTN based formulation, which was also an updated version of the STN of Ierapetritou and Floudas (1998) to address FIS without considering storage as a separate task. The model of Ierapetritou and Floudas (1998) was incapable of handling problems with dedicated finite storage due to the unit specific event point representation as already explained. In Shaik and Floudas (2008), new equations were proposed to ensure that for materials with FIS, \( r \in R^{FIS} \), zero wait was enforced for the tasks that produced or consumed the material in different units. In this way, the finishing time of the producing task in one unit was forced to coincide with the starting time of the consuming task in the other unit, thereby, avoiding real time violations associated with storage.

Susarla et al. (2010) modified the scheduling formulation of Sundaramoorthy and Karimi (2005) and employed asynchronous slots instead. The model allowed non-simultaneous transfer of material into and out of a batch process. This flexibility led to better schedules compared to existing models. Moreover, this formulation allowed tasks to span over multiple slots. The authors revealed this concept as a prerequisite for the validity of resource balances in multi-grid formulations for the following reason: If a unit was receiving or delivering material to a storage vessel at any time, the formulation forced the slots corresponding to both the unit and storage at that time to have the same index (unit slots were synchronized only across units that were associated with the same material). The ability of tasks to span multiple slots created the opportunity to insert dummy slots freely to any units when required. This enhanced the opportunity for units and storage tanks associated with the same material to have the same slot index. This also established the ability to register material transfers to storage chronologically.
Li et al. (2010) studied the performance of the most recent unit specific event based models of Floudas and co-workers. The conclusion of their investigation showed that not allowing tasks to span over multiple event points, might lead to suboptimal results for certain problems. The models of Shaik and Floudas (2007, 2008), exhibited such behaviour. The authors further buttressed the superiority of the unit specific formulations of Shaik and Floudas (2009) for event based models and Susarla et al. (2010) for slot based models.

2.2.3.3. SSN scheduling formulations

Majozi and Zhu (2001) introduced the SSN representation which considered states only. A single binary variable, $y(s, p)$ was defined which had a value of 1 if a state $s$ was used at time point $p$. The total number of binary variables associated with this representation was $i \times p$ a number less than that resulting from the formulation by Ierapetritou and Floudas (1998). When more than one state was used simultaneously in a unit, only one state was assigned the binary variable and was termed the effective state, $s^*$. This unit specific event point formulation led to fewer binary variables compared with other approaches of the time. No reduction in the number of binary variables was necessary as the correspondence between states, tasks and units was already taken into account. However, similar to Ierapetritou and Floudas (1998), this work was incapable of correctly addressing FIS policy.

Seid and Majozi (2012) developed a formulation to address the limitations of Majozi and Zhu (2001). The constraint proposed to handle dedicated FIS was different from that of Shaik and Floudas (2008). In Seid and Majozi (2012), the finishing time of a task producing an intermediate was not forced to coincide with the starting time of a consuming task. The unit in which the producing task occurred was allowed to store the intermediate material until the consuming task was ready to begin. This feature, the ability to store intermediate material in a unit that produced it offered some flexibility to problems with FIS and contributed to the improvement in objective values observed in some case studies when compared with the most competitive contemporary models: the STN, unit specific event based formulation of Shaik and Floudas (2009) and the RTN, asynchronous slot based formulation of Susarla et al. (2010). This formulation was shown to result in a reduction of computational time and led to fewer time points in some case studies. In addition, the formulation was capable of solving...
scheduling problems for long time horizons without resorting to the cyclic scheduling techniques prescribed by Wu and Ierapetritou (2004).

2.2.4. Scheduling formulations addressing changeover

Some of the scheduling formulations discussed above ignored changeover, e.g., Seid and Majozi (2012), Sundaramoorthy and Karimi (2005) and Susarla et al. (2010). Sequence independent changeover obtains in situations where batches being processed have similarities such as equipment set up and operating conditions. Sequence independent changeover time can be incorporated easily in the formulations just mentioned, by the addition of changeover time to the finishing times of tasks. Sequence dependent changeover on the other hand applies in situations where the equipment unit requires preparation in order to process different batches. During changeover, the equipment unit is not in operation. If changeovers are sequence dependent, the utilisation of equipment capacity will depend on the sequence of production. Castro et al. (2008) concluded that too many changeovers of sufficient magnitude could result in a significant reduction in the capacity available for production. This provided an incentive to develop scheduling models that accounted for sequence dependent changeover time and costs.

The discrete time formulations of Doganis and Sarimveis (2007), Kelly and Zyngier (2007) and Kilic et al. (2011) easily incorporated sequence dependent changeovers as a result of the common time grid for all units. However, a finer time discretisation was needed to account for small changeover times leading to large model sizes and excessive computational times. On the other hand, continuous time formulations cater for sequence dependent changeover without the need for prohibitive time discretisation but at the expense of larger integrality gaps. The subsequent reviews focus only on continuous time formulations addressing sequence dependent changeover for this reason.

2.2.4.1. Precedence based formulations addressing sequence dependent changeover

In Cerda et al. (1997), a binary variable was defined, which indicated that an order was the first to be processed in a unit. A second tri-index binary variable identified which order between a pair of orders was processed first in a unit. Combinations of these binary variables
captured an orders’ immediate successor or predecessor. The model was restricted to objective functions involving time such as the minimisation of tardiness and the minimisation of makespan. To reduce the size of the MILP, a heuristic ensuring that orders were processed in a sequence concomitant with their due dates was applied. Hui and Gupta (2001) improved on this formulation by reducing the overall number of binary variables and thus the solution time. The authors proposed two bi-index variables to handle order sequence dependence instead of the tri index binary variable. The formulation also depended on the use of similar heuristics and was restricted to similar objective functions involving time.

In the formulation of Mendez et al. (2001), only one binary variable was defined for sequencing a pair of orders. This constituted a general/global precedence relationship. Savings in the total number of binary variables was also achieved through the use of similar heuristics as in Cerda et al. (1997). A further reduction in computational effort was accomplished by reducing the total number of constraints through the manipulation and substitution of constraints.

The drawback of global precedence based models is that they are incapable of addressing changeover costs due to the fact that the exact successor of a task is unknown. The global sequencing binary variable is active for all pairs of batches assigned to the same unit regardless of the distance between them. Gupta and Karimi (2003) introduced local/immediate precedence to address this. Three binary variables were defined; the assignment binary variable, indicating that an order was processed on a unit, the binary variable indicating if an order was the first in a unit and the sequencing binary variable stipulating if an order was processed after another on a unit. Constraints involving combinations of these binary variables ensured that an order was either the first in a unit or followed another order processed on that unit. As such, it captured local/immediate precedence. The authors, however, limited their objectives to time considerations.

In the aforementioned methods, the batch sizes for the entire process were defined \textit{a priori} by assuming batching was complete. Sundaramoorthy and Maravelias (2008) overcame this shortcoming with a precedence based formulation for simultaneous batching and scheduling of multiproduct plants. The formulation included local sequence binary variables that were active only for adjacent batches assigned to the same unit, in an approach similar to Gupta and Karimi (2003). Objectives in this work included profit maximisation, where changeover
costs were minimised. Kopanos et al. (2009) also addressed profit maximisation using both local and global sequencing binary variables. The authors featured an auxiliary variable that determined which batch in a pair of batches was processed first as opposed to defining a binary variable that determined if a batch was first to be processed on a unit. Capon-Garcia et al. (2011) extended the immediate precedence model of Gupta and Karimi (2003) to address in addition to plant productivity, the environmental impact of the production process. The availability of more than one cleaning method for changeover purposes was considered. A sequencing binary variable, $X_{i,c}$ was 1 if batch $i'$ was processed immediately after $i$ and the unit was prepared using cleaning method $c$. As a result of the multiobjective nature of the problem, a compromise solution was selected from the set of Pareto solutions obtained. This strategy however, required high computational effort. To counter this, Capon-Garcia et al. (2013) proposed a hybrid strategy involving a combination of genetic algorithm and rigorous local search to solve the problem in shorter computational times compared to their previous work. The tradeoff however was that global optimality was not guaranteed.

2.2.4.2. Network based formulations addressing sequence dependent changeover

Maravelias and Grossmann (2003) incorporated sequence dependent changeover time into the sequencing constraint for different tasks in the same unit. Although changeover costs were not considered in the objective, the authors proposed the definition of a binary variable, $Y_{i,c,n}$ which was active if task $i$ performed on some unit, was followed by another task $i'$ at time point $n$. The constraint activating this binary variable related the finishing of task $i$ at time point $n-1$ with the starting of task $i'$ at time point $n$. It is evident that this formulation is unable to handle situations where a consecutive task in the same unit does not take place at the next time point but occurs at some other time point after that.

Janak et al. (2004) included sequence dependent changeover time in the constraint addressing the sequence of different tasks in the same unit. The formulation was based on STN and unit specific event point representation and performed better computationally than that of Maravelias and Grossmann (2003). Changeover costs were not considered. In another work, Shaik and Floudas (2008) considered sequence dependent changeover time in a formulation based on the resource task network, RTN, representation and unit specific event point
representation. In addition, the sequence constraint addressing different tasks \((i\text{ and } i')\) performed in the same unit contained big-M terms to effectively cater for the possibility of an intermediate task \(i''\) being processed in the unit after task \(i\) was completed, but before task \(i'\) took place in the same unit. Changeover costs were also not considered in this work.

In one of the RTN, multiple time grid formulations developed by Castro et al. (2006), sequence dependent changeover was considered as an explicit task by combining a processing task and the corresponding changeover task into a single task. This served to reduce the total number of model variables as a single time interval was sufficient for each combined task instead of an interval each for processing and changeover. Excess resource variables were used to identify the cleaning state of a unit. A unit could execute a combined task only if it was in the right cleaning state. There was an initial cleaning state before a task was performed and a corresponding cleaning state after the task was completed. The formulation could address objective functions such as the minimization of total cost and the minimization of makespan. A drawback of this formulation was its large size. The formulation was modified by Castro et al. (2008) to require fewer time points and yield smaller models. In this work, all batches of the same product processed in the same unit were aggregated into a single task (called explicit batching). The aggregate task duration included the time to produce all selected batches added to the total changeover between different batches of the same product and the changeover required to prepare the unit for the next aggregate task. Although the formulation resulted in faster computational times and could solve large-scale problems, it was unable to handle variable batch durations due to the aggregation of tasks. Furthermore, changeover costs could not be computed.

Thus far, the discussed network structure formulations did not address sequence dependent changeover costs explicitly. When profit was maximized (total cost minimized), the objective was to produce products that were the most valuable and less time consuming, while meeting their minimum demands. The prerequisite to account for sequence dependent changeover costs is the ability to correctly establish the immediate successors of all tasks in each unit. This is not a trivial problem when one considers that although tasks occur consecutively in the same unit, they do not necessarily take place at consecutive time points/slots. That is, the difference between the time points/time slots of two consecutive tasks on a unit may be greater than 1.
Erdirik-Dogan and Grossmann (2008) addressed sequence dependent changeover costs in their MILP model for multistage, multiproduct batch plants based on asynchronized slots. Changeovers within time periods and across time periods were determined. The authors defined within each period, a changeover binary variable \( Z_{i,l,m,i',l',t} \), which was activated if product \( i \) assigned to slot \( l \) was followed by product \( i' \) at slot \( l+1 \) on unit \( m \) at time period \( t \). This was sufficient as the formulation was modelled such that any unused slots within a time period were placed as the last slots in the time period. A similar binary variable was defined to account for changeover across time periods. Due to the possibility of the existence of empty slots when transitioning from one time period to the next, another binary variable was defined to determine whether a slot was used or not. Constraints with combinations of these binary variables determined the last utilized slot of each time period. This way, the immediate successor of any batch in a unit was correctly identified. The total changeover cost was computed from the summation of the product of all active changeover binary variables and the relevant changeover costs. A decomposition algorithm was employed to improve computational times.

Gimenez et al. (2009) employed the concept of unit states/modes similar to the clean states of Castro et al. (2006), to capture the status of a processing unit. The set of possible modes for a processing unit were: clean (ready for the execution of any processing task), clean\(_i\) (ready for the execution of a processing task \( i \)) and dirty\(_i\) (dirty after the execution of task \( i \)). A task performed in a unit consumed a mode at the start and produced a different or the same mode at the end. Changeover was required when, after a task was completed, the mode of the unit was incompatible for the next task. In this work, changeover was treated as a task characterized by a fixed duration and a fictitious batch size equal to 1. This way, the definition of an assignment binary variable was sufficient to capture a changeover task occurring in a unit at a time point. Additional variables and constraints described the consumption and production of the different modes. Changeover costs were computed through the summation of the product of all active assignment binary variables associated with changeover tasks and the corresponding changeover costs. The formulation of Gimenez et al. (2009) was based on the RTN representation and employed global time points.

Recently, Kabra et al. (2013) addressed changeover costs by extending the model of Shaik and Floudas (2007) and adapting the changeover constraints of Shaik et al. (2009). To
account for changeover costs, a [0, 1] continuous variable, $wc(i, s', s, n')$ was introduced which assumed a value of 1 when state $s'$ processed by task $i$ at event $n'$ was succeeded by state $s$ at any later event $n$. Additional constraints were described to address the possibility of empty events between consecutive tasks and to ensure that when the changeover variable was active for a pair of tasks, no intermediate tasks existed between them. This way, the unique successors of tasks could correctly be identified and changeover costs were determined from the summation of the product of all active changeover variables and the corresponding costs.

2.3. Process integration

Process integration originated from the concept of heat integration in the processing industry, in response to the energy crisis of the 1970’s when energy efficiency became a priority.

2.3.1. What is ‘Pinch Analysis’?

Linnhoff and Flower (1978) developed the seminal ‘Pinch Analysis’, whereby the hot and cold utility requirements of a system of processes were determined after evaluating the existing streams of the system for possible heat integration. On a plot of temperature versus heat load, composite curves of the system were generated which provided a visual representation of the overall heating and cooling demands of the processes. An illustration is provided in Figure 2.2. Other insights obtainable from the graph were maximum heat integration possible and the additional heating or cooling requirements to be met with external hot and cold utilities respectively. Expanding on this, a grand composite curve (GCC) was derived to further illuminate the pinch point, the point at which a minimum temperature difference existed. On the GCC, the pinch point divided the system into two distinct parts; above the pinch, a heat deficit region and below the pinch, a heat surplus region. The rules of heat exchanger placement were that heat should not be transferred across the pinch point and equipment releasing heat should be placed above the pinch while equipment requiring heat should be placed below the pinch.
Different rules for the placement of other equipment with respect to the pinch have since been developed (Linnhoff et al. (1983) for distillation columns; Townsend and Linnhoff, (1983) for heat pump and heat engines etc.,) and the rules for transfer of heat across the pinch have been relaxed and shown to be beneficial to the system (Bagajewicz and Barbaro, 2003).

The advantage of pinch analysis is that it provides performance targets ahead of the synthesis of heat exchange networks.

![Composite curves showing heat integration](image)

Figure 2.2. Composite curves showing heat integration

### 2.3.2. Application of pinch analysis to other areas

Process integration has since grown to include a broad spectrum of methodologies with the objective of reducing the consumption of valuable resources and the emission of harmful substances into the environment, by combining several parts of process operations (Klemes and Kravanja, 2013).

El-Halwagi and Manousiouthakis (1989) adopted the principles of Linnhoff and Flower (1978) and introduced the Mass Pinch for mass exchange networks, where concentration was the driving force instead of temperature. Mass integration involved the transfer of mass from rich process streams to lean process streams while satisfying the target outlet concentrations with the aim of identifying performance targets and optimising the flow of mass throughout
the process. This was applicable whenever process streams were exchanging mass in a number of transfer units, e.g., absorbers and extractors.

The Water Pinch was developed by Wang and Smith (1994) who extended the general Mass Pinch concept to water usage. The objective was to target and synthesize for maximum water reuse and reduce wastewater effluent with an associated reduction in freshwater demand. A limiting composite curve was generated from a plot of concentration against mass load for a system of water using processes. Against this curve, a freshwater supply line was matched and the inverse of the slope of this line gave freshwater and wastewater targets of the overall process. An illustration is provided in Figure 2.3. In the figure, the limiting composite curve and the water supply line intersect at two points, coordinate \((0, 0)\) and at the pinch point. The meaning of the pinch point here is different from the pinch point of heat integration. It does not imply zero concentration difference. At this pinch point, the concentration of water is at its maximum value, at other points excluding zero concentration the concentration of water is below the maximum. The freshwater and wastewater targets were equal because there was no loss or gain of water. Next, the water network corresponding to the determined target was synthesized using a method analogous to the heat exchanger network design by Linnhoff and Hindmarsh (1983).

![Figure 2.3. Limiting composite curve (Wang and Smith, 1994)](image-url)
2.3.3. **Process integration techniques for energy and water in literature**

In literature, process integration techniques addressing energy and water as individual problems converged into two main categories, as methods employing heuristics became more redundant. In the first category, the techniques relied on pinch analysis. By applying thermodynamic and mass transfer concepts creatively, authors generated novel ideas to investigate a wide variety of problems. The second category of techniques relied on mathematical programming, usually based on superstructures which considered all the possible flow configurations the authors were interested in. Here, ideas generated from pinch analysis and more advanced features were formulated mathematically and the spectrum of process integration problems further increased in variety and complexity. This is because mathematical programming affords the opportunity to simultaneously determine trade-offs involving factors such as raw material costs, operating costs, product revenue, investment costs etc., for truly integrated solutions (Klemes and Kravanja 2012). In addition, it is useful in circumstances difficult to represent graphically. With mathematical programming, process integration is generally performed simultaneously with network synthesis. Shortcomings of mathematical programming are the difficulty associated with problem formulation and computer technology constraints.


Research addressing the same topics in batch processes generally got its direction from ideas generated from continuous processes. Although existing literature is not as vast, the continued emerging popularity of batch processes for industrial application has stimulated investigation. In batch processes, added to the constraints of temperature driving forces and concentration in energy and water optimisation problems respectively, is the time constraint. Without considering the timing of batch processes, results obtained from the direct application of techniques developed for continuous processes are usually infeasible in practice. Thus, time constraints make the integration of batch processes more difficult than
continuous process integration. The following sections are reviews on techniques for heat integration and water optimisation in batch processes.

2.3.4. Heat integration in batch plants

Heat integration involves the matching of streams requiring cooling with streams requiring heating in order to minimise the use of external utilities. Heat transfer from a hot source to a cold sink is constrained by both temperature and time in batch processes. The temperature constraint dictates that heat cannot be transferred from a hot stream to a cold stream if the temperature of the hot stream is lower than that of the cold stream. The time constraint dictates that a hot stream cannot transfer heat directly to a cold stream if the hot stream exists at a later point in time than the cold stream (Wang and Smith 1995). Indirect heat exchange through heat storage can be appropriate when the hot and cold streams are offset by some duration of time.

2.3.4.1. Earliest heat integration techniques for batch processes

The earliest studies investigating heat integration problems in batch processes resorted to pinch analysis and the use of heuristics. These studies generally extended the methods developed for heat integration in continuous processes.

Clayton (1986) presented the Time Average Model (TAM) in which, the discontinuous behavior of batch processes was ignored by assuming that hot and cold streams existed simultaneously, like in continuous processes. The heat load of each stream was averaged over the batch cycle time and the heat integration potential of the process streams and the minimum targets for external cooling and heating were then determined using pinch analysis.

Obeng and Ashton (1988) presented the Time Slice Model (TSM), an improvement of the TAM. In the TSM, the intermittent existence of streams was considered. The batch cycle was subdivided into time intervals (slices). The boundaries of the time intervals were set by the starting or ending of a stream (points at which change in heat flows occurred). Pinch analysis was then applied in these time slices. For every time slice, a local pinch temperature was
determined. The minimum utility required for the entire batch cycle was the combination of the minimums for each time slice.

Another improvement over the TAM was the time-dependent, heat cascade analysis for batch processes developed by Kemp and Macdonald (1987, 1988). The process was divided into time intervals and heat cascades calculated for each slice in a manner analogous to temperature intervals in the problem table analysis for continuous processes (Linnhoff et al. 1982). These cascades highlight for each time slice, the maximum internal heat exchange possible and the minimum utility requirement. A time energy cascade was then constructed to determine the energy targets. The analogous composite and grand composite curves were graphical representations of the heat cascades and were useful for the synthesis of heat exchange networks. Kemp and Deakin (1989) continued with cascade analysis for maximum heat recovery that considered both direct and indirect heat integration and introduced a 3-D plot to aid visualisation of heat flows. The authors also highlighted that rescheduling was a useful strategy that could lead to higher flexibility, reduced energy targets and increased plant capacity.

Vaselenak et al. (1986) were some of the first authors to employ a mathematical formulation for the investigation of heat integration. The authors considered the heat integration of vessels containing hot and cold fluids requiring cooling and heating respectively over a given production schedule. Co-current heat exchange, counter-current heat exchange or their combination were investigated. In this work, the time schedule was not accounted for. The authors developed a MILP and a heuristic procedure to determine the best pairing of hot and cold tanks for maximum heat recovery assuming all tanks to be available at the same time. When the final temperature of the tank was limiting, the MILP was adopted, conversely, when the final temperature was not limiting, the heuristic procedure was adopted.

2.3.4.2. Techniques for heat integration in batch plants after the 80’s

Methods addressing heat integration in batch plants can be classified based on the handling of the timings of batch processes as fixed schedule, where time is fixed or variable schedule, where time is a variable to be optimised.
Graphical/conceptual techniques relying on a fixed schedule

Kemp and Macdonald (1988) pointed out that although the TAM identified the energy integration potential, the obtained energy targets could not be achieved realistically through direct integration only, due to the fact that the time schedule was not considered. Stolze et al. (1995) addressed this limitation by means of heat storage. After establishing energy saving potential with the TAM, the authors applied a simple combinatorial method to determine the optimum number of heat storage units that was not economically prohibitive.

In Wang and Smith (1995), a plot of heat transferred against time called the energy composite curve was devised to visually recognise opportunities for both direct and indirect heat integration and determine the minimum energy target. Time was treated as the primary constraint and temperature feasibility as the secondary constraint.

In the method of Uhlenbruck et al. (2000) all possible heat exchanger networks featuring direct heat exchange were synthesized without analysing the stream matches from a thermodynamic perspective first. A given schedule was divided into time and temperature intervals. In order to apply heat cascade analysis, steady state heating and cooling within the time intervals was assumed. The overall utility targets were the sum of the individual targets of each time interval. One-to-one heat exchange between a hot and cold stream through a countercurrent heat exchanger was permissible. Further heat recovery was achieved through the inclusion of matches between residual and previously unmatched streams. The thermodynamic optimum could not be achieved by this method.

Krummenacher and Favrat (2001) developed a graphical pinch analysis based technique to determine the minimum number of heat storage units and their feasible operating range for heat recovery to minimise utility. Direct heat integration was not considered to avoid compromising the flexibility of the plant. The authors proposed a number of heuristic rules to investigate solution options corresponding to minimum costs. Rescheduling was also considered with the goal of decreasing storage capacity. A plot of total annual costs versus heat recovery assisted with demonstrating the trade-offs associated with storage supply temperatures. The authors also provided guidelines to extend the method to mixed direct and indirect heat integration.
Pourali et al. (2006) combined pinch analysis and an energy storage system with time decomposition in their approach. The method involved the designation of a number of time intervals and the identification of different combinations of these time intervals. The basis for this work was that the intervals may have better integration opportunities if combined, combinations such as singular, binary, ternary and higher. The problem table algorithm was applied to each combination to determine the hot and cold utility targets. In addition to energy considerations, the effect on costs of the combined intervals was investigated. Plant rescheduling was performed to match the optimum cost and plant operation for better heat integration. The disadvantage of the method was that the number of possible time interval combinations could be exhaustive. To counteract this, the authors recommended a computer program algorithm to simulate solutions for the different time interval combinations.

Foo et al. (2008) addressed the issue of minimum heat exchanger units within the batch HEN by adapting their earlier work for batch MEN (Foo et al., 2004, 2005). The approach which involved time dependent cascade analysis also took into consideration the possibility of common heat exchangers, that is, heat exchangers that were used in more than one time interval. The authors showed that ignoring these common heat exchangers led to suboptimal results, that is, more than necessary heat exchangers.

Recently, Dowidat et al. (2014) adopted the TSM and demonstrated that by introducing additional time slices, the prediction of heat integration potential could be facilitated without loss in accuracy. This was as a result of transforming dynamic heat streams, streams for which the temperature difference between the medium and the utility was time dependent, into steady state streams. The intersection points of the temperature-time curves of coexisting cold and hot streams divided the dynamic streams into sub-streams. The TSM could then be applied at all time intervals and heat integration targets could be identified in each interval. The average temperature of a dynamic stream within an interval was used as the temperature for analysis in the TSM. Direct heat integration was the focus of this work. As would be expected, it was found that when compared to a true steady state stream, the maximum internal heat exchangeable for dynamic streams were lower for the same data.
**Mathematical techniques relying on a fixed schedule**

Vaklieva-Bancheva *et al.* (1996) developed an MILP mathematical formulation to investigate direct heat integration for minimizing total costs (utility consumption). Only specific pairs of hot and cold vessels which required heating and cooling respectively were allowed to be heat integrated. A drawback of this work was that the authors only addressed the special case of a plant operated in a zero-wait overlapping mode, in which each product passes through a subset of equipment stages and production takes place in campaigns.

Ivanov *et al.* (1993a, b) investigated heat integration between a hot and a cold reactor which were active at different time intervals with another objective of analysing the arrangement of a heat storage system. A two storage tank system and one combined heat storage system were considered. The two storage tank system, one hot and one cold, was considered for situations where the number of hot and cold reactors was equal. The combined heat storage system was considered for situations where the number of hot and cold vessels was unequal. For each storage system, three different arrangements were investigated and mathematical models were developed to describe the temperature variations within the vessels with time. The three cases were: both hot and cold streams were recycled, one stream was recycled and both streams transferred to receiving tanks. Heat losses were assumed negligible and flowrates of streams were assumed constant. Ivanov *et al.* (1993c) applied the results from their works (Ivanov *et al.*, 1993a, b) to synthesize new heat integrated plants or retrofit existing ones using heat storage. The mathematical models obtained were nonlinear, for which adaptive nonlinear optimisation was employed to solve.

Corominas *et al.* (1994) developed a formulation applicable to multiproduct plants operated in campaign mode, for which the coincidence of hot and cold streams was less frequent due to product changeovers (high variety of products.). To satisfy the objective of designing a HEN for minimum energy consumption, rescheduling was performed to ensure maximum overlapping times between hot and cold streams of a pre-specified campaign for possible heat integration. The mathematical formulation made allowances for task delays and used heuristic procedures to shift the relative timings of hot and cold operations wherever necessary. Only direct heat integration and one to one stream matches were considered.
Chen and Ciou (2008) proposed a mathematical model for the simultaneous analysis of an indirect heat exchanger network and its targets. The associated thermal storage policy for minimising external utility was also determined. The formulation resulted in an MINLP, solved with a global solver. The effect of multiple energy storage vessels was investigated and the authors found that increasing the number of vessels did not guarantee that heat recovery would improve. Later, Chen and Ciou (2009) relaxed the constraint which fixed the storage temperature of the recirculating heat transfer medium to increase heat recovery potential.

**Mathematical techniques relying on a variable schedule**

The methods relying on a fixed schedule assume that the given schedule is optimal. According to Papageorgiou et al. (1994), a large number of different schedules often exist which achieve the same optimum objective value that satisfies the production requirements. The potential for heat integration can vary considerably from one alternative optimal schedule to the next. Therefore, applying heat integration to a fixed schedule may result in low heat recovery, and hence, simultaneous optimisation of both scheduling and heat integration is best. In this category of methods, time is not a given parameter but an optimisation variable and the methods are supported by the vast improvements made to batch scheduling formulations in terms of robustness and reasonable computational times.

Papageorgiou et al. (1994) presented one of the earliest attempts for the simultaneous optimisation of batch scheduling and heat integration. In this work, the discrete time, STN scheduling formulation of Kondili et al. (1993) was adopted and additional binary variables and heat balance constraints were included to capture direct and indirect heat transfer. Whether tasks were to be heat-integrated or remained in standalone mode was predefined. The authors recommended that the cost of utilities be included during the maximisation of production over a given time horizon. In addition, possible heat losses from the storage tank were also considered. The developed model was a nonconvex MINLP which made the model complex and computationally difficult to solve.

Lee and Reklaitis (1995a) developed an MILP mathematical formulation that investigated direct heat integration in a single product, cyclically operated batch plant with NIS policy. Heat exchange times were considered negligible compared to batch processing times and
took place between the streams as they were transferred from unit to unit. In this work, only one-to-one matches were considered with the objective of determining the operating schedule for maximum heat integration between streams. Multiple heat exchange modes in the form of concurrent, countercurrent or a combination of the two were analysed. Significant savings in utility costs were obtained in conjunction with the rescheduling of operation times. Later, the authors considered nonnegligible heat transfer times and shared heat exchange units across multiple matches in their formulation (Lee and Reklaitis, 1995b).

The aforementioned mathematical formulations were restricted to one to one stream matching between hot and cold streams for heat integration. The possibility of multiple stream matching could present a greater degree of heat recovery. In the MINLP formulation of Zhao, et al. (1998a, b) which was based on cascade analysis, multiple stream matching was allowed in which the supply and target temperatures of all streams determined the temperature intervals for heat exchange. The mathematical formulation was based on the scheduling framework of Papoulia and Grossmann (1985) and considered cyclically operated batch plants with NIS. Model preprocessing was proposed to reduce the size of the model involving the solution of a series of MILP models. The formulation was restricted to the solution of small problems.

In the approaches so far, the performance of equipment was assumed to be constant over time and the effects of fouling on heat integration were ignored. Fouling causes the performance of heat integrated operations to decrease as resistance to heat transfer increases with the deposition of fluid over time. Georgiadis and Papageorgiou (2001) considered heat integration in multipurpose batch plants together with the effects of fouling and the necessary cleaning required. Frequent cleaning could lead to losses in production time and insufficient cleaning could lead to decreased energy efficiency. The discrete time, STN formulation of Kondili et al. (1993) was the scheduling framework. The model determined the production schedule, the number of cleanings and their timings, the utilisation of external utilities and the flowrates of the heat transfer medium over time. Heat integration between units was indirect, removing the need for synchronisation between the tasks producing and consuming heat. Heat losses to the environment from the heat storage tank were taken into account. The overall formulation was an MINLP for which an iterative procedure involving the solution of a series of MILP and NLP subproblems was employed. The method, however, could not guarantee that the solution was optimal or even feasible.
Solving mathematical models associated with scheduling problems especially given the computational capabilities of the time was difficult. The inclusion of heat integration as an additional component increased the mathematical complexity significantly. Adonyi et al. (2003) proposed their graph theoretical approach, S-graph to address this. The original S-graph framework was designed for the NIS policy with an objective of determining a schedule with minimal makespan (Sanmarti et al., 2002). As already mentioned, such a solution may not be flexible enough to allow for heat integration between tasks. Adonyi et al. (2003) combined heat integration and scheduling to determine a solution that satisfied a makespan constraint and minimal utility as a goal. The makespan constraint was in the form of an upper bound on the makespan. Results showed that reduction in utility consumption was possible when the makespan was increased slightly.

The mathematical formulation of Majozi (2006a) avoided the disadvantages of discrete time formulations by employing the continuous time, SSN, scheduling framework of Majozi and Zhu (2001) and investigated direct heat integration in multiproduct batch plants. Two scenarios were explored. In the first scenario, the energy requirement was dependent on batch size which could vary and the second was based on fixed batch sizes for which, the heat requirement was a given parameter. The first scenario resulted in a MINLP, linearised exactly to an MILP problem using Glover (1975) transformations. The second was an MILP problem. For plant operability purposes, only one to one heat integration between units was allowed. In this work, the assumption was made that the heat driving forces and heat duties between heat integrated operations were sufficient. As a result, throughput was unaffected by heat integration.

Majozi (2009a) extended this work to consider indirect heat integration through heat storage and significant savings in external utility consumption were observed. The capacity and initial temperature of the heat storage vessel were predefined and the overall model was an MILP, the solution of which is guaranteed to be the global optimum. Stamp and Majozi (2011) extended the formulation even further to address the design of the heat storage tank and the effect of heat losses from the heat storage vessel. The formulation was applicable to multipurpose facilities and was an MINLP, which was linearised using the Glover techniques (Glover, 1975) and the reformulation-linearisation technique of Quesada and Grossmann (1995).
Chapter 2. Literature Review

Chen and Chang (2009) addressed both short term and periodic scheduling in their formulation in an effort to generalize the heat integration model of Majozi (2006). Also in comparison, this formulation did not restrict heat integration between tasks to their starting times. The start of heat integration could be shifted from the beginning of a task using an adjustable parameter, a useful feature for practical purposes when preheating was required before heat integration. The authors employed the continuous time, RTN formulation of Castro et al. (2003, 2004) and considered only direct heat integration. Only one to one integration was considered and the overall model was an MILP.

The difficulty when solving mathematical formulations for heat integration in batch plants becomes much more obvious when applied to complex real life problems. Consequently, Halim and Srinivasan (2009) developed a sequential technique as opposed to a simultaneous one. The method exploited the existence of alternative optimal batch schedules. The batch production schedule that satisfied a given production objective e.g. maximum profit or minimum makespan was optimised first and a number of optimal schedules were generated. The scheduling formulation employed was the STN, synchronous time slot formulation of Sundaramoorthy and Karimi (2005). Heat integration using TAM and TSM was investigated for the generated optimal schedules and the schedule with the minimum utility targets was selected.

Tokos et al. (2010) presented a MILP mathematical formulation to study heat integration between batch operations for the specific industrial conditions of a beverage plant. The heat integration mathematical formulation of Lee and Reklaitis (1995a,b) was employed. Co-current heat exchange was assumed as batches remained in their processing units during heat exchange (in situ). The original model of Lee and Reklaitis (1995a, b) only catered for one-to-one matching but this work considered multiple matching between several units. The objective function included the annual utility costs savings and annual investment costs of heat exchangers to investigate the economic trade-off between utility savings and investment costs. Significant savings in utility were observed from this investigation.

The above mentioned works addressing heat integration of batch plants have mainly considered direct or indirect heat integration between processing units. Very recently, Lee, Seid and Majozi (2015) investigated heat exchange between process streams during material transfer. The mathematical formulation took advantage of the intermittent continuous
behavior of process streams during material transfer from one equipment unit to another with the objective of maximising the occurrence of coinciding hot and cold stream pairs that satisfy temperature driving forces. The options available for heat integration in the superstructure presented were between two (one hot, one cold) output intermediate streams or between an output intermediate stream and a raw material stream. A one-to-one heat integration arrangement was assumed for practicality. Heating and cooling through external utilities occurred \textit{in situ}. Compared to heating and cooling \textit{in situ}, heat integration during stream transfer had the advantage of shortening process time, due to the dedicated time required for heating or cooling in a processing unit. In their investigations, utility requirements were reduced and throughput was improved. The mathematical model was an MINLP solved with a superior MINLP solver, BARON in GAMS.

Heat integration has found successful practical application in the processing (petrochemical, chemical, food) and power generating industry over the last 40 years (Klemes and Kravanja, 2012). Literature examples are: production of oleic acid from palm olein (Foo et al., 2008), a beverage plant (Tokos et al., 2010), a milk powder plant (Atkins et al. 2010) and a cheese factory (Becker et al. 2012). Heat integration is still a thriving research area.

2.3.4. Wastewater minimisation in batch plants

Process integration techniques have been widely adopted as a reliable tool to minimise water. These techniques capitalise on the potential to reuse water with certain levels of contamination. Consequently, both freshwater consumption and wastewater generation are reduced when water reuse opportunities are maximised. Water users have been classified either as fixed load or fixed flow operations. In fixed load operations, water is modelled as a mass transfer agent, MSA into which contaminants of a given mass are transferred. Fixed flow operations are characterised as water sources and sinks that produce and consume a fixed amount of water respectively. In batch plants specifically, water reuse is limited by both the suitability of contaminated water for each water using operation and the timing of operations. Techniques for wastewater minimisation have sought to overcome these limitations and can be broadly divided into insight-based and mathematical techniques.
2.3.4.1. Insight based techniques for wastewater minimisation

Insight based techniques revolve around pinch analysis developed for continuous processes and are either graphical or algebraic. These methods involve the determination of the minimum freshwater target, which provides insights such as the identification of bottlenecks during network synthesis. However, these techniques are restricted to problems with single contaminant and fixed time schedules.

In the work of Wang and Smith (1995), the problem data was divided into concentration intervals and time subintervals, that is, concentration was treated as the primary constraint and time as the secondary constraint. Targeting was performed in each concentration interval by reusing water which was available, in its own time subinterval where possible. Any surplus water could be reused in subsequent time intervals in the same concentration interval or could be stored for reuse in subsequent concentration intervals. When no opportunities for recycle/reuse existed, freshwater was used. This method permitted water reuse between units that were still in operation with overlapping time intervals. This is possible for processes operating in a semicontinuous mode but not possible for strictly batch operations, where water is available from a unit only at the end of its operation and can be transferred to a unit only at the beginning of its operation.

Majozi et al. (2006) modified the technique of Wang and Smith (1995) to address strictly batch processes. Two scenarios were considered to determine the effect of the targeting procedure on the final design. In the first scenario, similar to Wang and Smith (1995), time was the primary constraint and concentration the secondary constraint. In the second scenario, the primary constraint was concentration and the secondary constraint was time. In the second scenario, targeting was performed in each time interval and whenever possible, any surplus water was transferred to higher concentration subintervals in the same time interval for reuse, or stored for indirect reuse in later time intervals but could not be reused in lower concentration subintervals or previous time intervals. Following this procedure, any shortfall within any concentration subinterval could be made up from freshwater. At the end of their analysis, the authors found that reversing the priority of time and concentration constraints did not have any effect on the water target. The choice of which constraint to use could be based on the ease with which the targeting could be performed. The methods by Wang and Smith (1995) and Majozi et al. (2006) were applicable to fixed load problems.
Foo et al. (2005) presented a numerical technique that was applicable to fixed flow problems. The water cascade analysis, WCA, developed for continuous processes (Manan et al., 2004) was utilised to determine water targets in batch processes. In the two-stage method, targeting to identify the minimum water flows was performed first using a time-dependent water cascade analysis and network synthesis was performed next for each time interval using the guidelines stipulated by Hallale (2002). The overall network called the time-water network diagram was then developed using information from the sub-networks of each time interval. From the time-water network diagram, the interaction between the water demands/sources and time could be clearly appreciated. The effect of a storage tank was also considered in this work. The storage capacity, defined as the maximum amount of water that was sent to the storage tank was determined. To prevent mixing water sources of different concentrations, the number of storage tanks was equal to the number of water sources at different concentration levels that were intended for reuse.

Chen and Lee (2008) introduced a batch network design technique using a graphical representation called the quantity-time diagram which allowed connections between sources and demands to be visualised. In this way, the utility consumption and network structure were determined simultaneously, unlike in aforementioned methods, where synthesis was performed after targeting. Completely batch and semi-continuous processes modeled as fixed flowrate problems could be handled by this method and storage was also considered to enhance water recovery. Care was taken when selecting water sources to be sent to the storage tank so that the concentration of water in storage did not exceed the maximum allowable requirement of the main water users. The number of storage tanks could be reduced by sending reusable sources with similar concentrations to a storage tank. At the end of the network design, the optimality of the results was checked using any of the established targeting methods in the literature. It was found that the design results by Chen and Lee (2008) agreed with the freshwater targets from pinch analysis. The drawback of their method was that it could require multiple iterations to determine the optimum design.

Liu et al. (2007a, b) developed an algebraic targeting technique called the time-dependent concentration interval analysis, CIA for fixed load and fixed flowrate problems. In this method, water sinks and sources were distributed in their respective time intervals and then targeting was performed using a concentration interval table, CIT. In summary, the CIT involved arranging impurity concentrations in descending order and locating water flows into
the appropriate concentration intervals. The impurity load in each concentration interval was calculated as the product of water flow and the concentration difference across the interval. Next, the impurity load was cascaded upwards throughout the concentration intervals. The minimum freshwater flow was determined by dividing the cumulative massload by its corresponding concentration level. This procedure was repeated for all time intervals to determine the overall minimum freshwater flow. A number of batch processing scenarios were investigated including a single batch operation with and without storage and cyclic batch operation with water storage. The method was applied to case studies involving semi-continuous processes.

The main contribution by Kim (2011) was the systematic design procedure for a water network in which wastewater was minimised. The first step involved determining a lower bound target, LBT from a limiting water flow composite curve using the method of Wang and Smith (1995). Next, the typical procedure of allocating water using operations to their appropriate time intervals was followed and targeting was performed within each time interval and summed over all intervals to obtain an upper bound target, UBT. The design of the network applied the water mains method of Kuo and Smith (1998) in conjunction with a few design rules. The resulting network did not necessarily coincide with the LBT. The UBT and LBT provided a range of possible minimum flow that could be achieved during network design. The method was applicable to fixed load problems that operated semicontinuously and catered for both single batch and cyclic batch operations.

The main contributions by Chaturvedi and Bandyopadhyay (2012) were the vigorous mathematical proofs that explained the effectiveness of the targeting procedure employed to obtain the minimum resource requirement. In this work, the authors proved that for a single batch operation, the minimum resource requirement could be determined through sequential transfer of waste from one time interval to the other. Where cyclic batch operation was concerned, the authors demonstrated that the minimum resource requirement could be targeted like a system of continuous processes by collapsing all time intervals into a single interval. Chaturvedi and Bandyopadhyay (2013) extended their previous work to consider multiple resources. The objective of this work was the minimisation of operating cost of the batch process but did not include cost of equipment, storage or piping units. Both contributions were applicable to fixed flow problems.
In general, graphical techniques are constrained to two dimensions hence; they cannot be used to solve problems with more than one contaminant effectively. It is important to note that the above insight based techniques required that the starting and finishing times of the water using operations be defined. This meant that the operating schedule could not be optimised. Once an operating schedule is given, the minimum water targets determined are unique to the schedule and so are not guaranteed to be the true minimum.

2.3.4.2. Mathematical techniques for wastewater minimisation

Wastewater minimisation using mathematical optimisation is of particular interest due to its suitability in addressing more complex batch problems involving multiple contaminants, elaborate objectives such as cost optimisation and additional constraints such as limited piping, forbidden connections etc. (Foo 2009, Gouws et al., 2010, Jezowski, 2010).

Techniques relying on fixed schedule

Almato et al. (1999) proposed a mathematical approach that considered indirect water reuse through storage tanks. This was to override the requirement that the starting time of the receiving process and the finishing time of the discharging process coincide for direct reuse to occur. In the model, direct reuse was not considered as an option. After optimisation, the fraction of effluent water sent from units to the storage tanks of known capacity and that sent from each tank to each process (tank stream assignment) was determined. The model derived for fixed load problems, was an NLP for which heuristics were applied to provide an initial starting point for the solution. Global optimality could not be guaranteed.

In the formulation by Kim and Smith (2004), both direct and indirect water reuse were considered. The model was derived for fixed load problems and considered multiple contaminants. The authors argued that the formulation of Almato et al. (1999) led to unnecessary mixing, thus reducing concentration driving force and opportunities for water reuse. The formulation of Kim and Smith (2004) assumed a storage tank for each unit generating wastewater, which could serve as reuse water to other units. The actual presence of such storage tanks was determined after optimisation. The objective was the minimisation of overall cost including freshwater costs, piping and storage costs. The model was an
MINLP for which a starting point was provided using an iteration scheme between an MILP and LP decomposition of the original model. The solution strategy could lead to long computational times.

Similar to Almato et al. (1999), no direct reuse was incorporated in the formulation by Shoaib et al. (2008). Water from sources could only be reused by water sinks in subsequent batch cycles to avoid violating time constraints. A three-stage hierarchical approach was employed for batch water network synthesis. In the first stage, the minimum freshwater and wastewater targets obtained by maximising reuse opportunities were determined. In the second stage, the minimum number of storage tanks was determined, subject to the aforementioned targets. The third stage involved the reduction of the number of interconnections between sources and sinks. In addition, this work examined the effect of a water regeneration unit for the further reduction of freshwater requirements. Of interest was the appropriate placement of the regeneration unit, which was governed by insights from pinch analysis. The method was derived for fixed flow problems with a single contaminant and the resulting MINLP which was relaxed to an MILP using McCormick (1976) transformations. The relaxed MILP provided a starting point for the MINLP.

Majozi (2005a) developed a formulation for fixed load problems with a single contaminant that considered direct reuse/recycle and circumvented the use of storage tanks. This required the inclusion of sequencing constraints to ensure that water reuse coincided with the finishing time of the water producing process and the starting time of the water consuming process. The bilinearities in the contaminant mass balances were linearised by fixing the outlet concentrations and making substitutions and Glover (1975) transformations. This way, the original MINLP was transformed to a MILP.

Further water reuse can be achieved through interplant integration in industrial complexes where, multiple plants grouped in different geographical locations exist. Chew et al. (2008) discussed two different schemes for interplant water integration, direct or indirect integration. In direct integration, water from different plants was integrated through the use of connecting pipelines. In indirect integration, a centralized hub collected water discharged from plants and served as a water source to other plants. Lovelady et al. (2009) extended the model of Chew et al. (2008) to address property integration and the synthesis of eco-industrial parks.
Lee et al. (2014a) investigated interplant network synthesis in systems consisting of both continuous and batch processes. All batch processes were treated as continuous through the use of two storage tanks, one at the inlet and the other at the outlet of a batch process to override time. The model could be solved as a two-stage mathematical model. The objective in the first stage was the minimisation of freshwater consumption. The second stage involved the minimisation of the total storage capacity subject to the predetermined freshwater target. The formulations of both stages could be combined and solved simultaneously to determine economic tradeoffs. The objective, in this case, was the minimisation of total network cost consisting of freshwater costs, wastewater disposal and the cost of piping and storage tank costs. The formulation was derived for fixed load processes.

In a separate work, Lee et al. (2014b) developed a mathematical model to minimize freshwater consumption, wastewater generation, the number of storage tanks and the number of interconnections for cyclic batch operations featuring both fixed load and fixed flowrate processes where multiple contaminants were present. The waste management hierarchy (WMH), of Wan Alwi et al. (2008) which systematically screened different wastewater minimisation options, was adopted. The procedure enabled planners to consider water reuse, recycling and regeneration as well as water source/sink elimination and reduction. The mathematical model was solved in four stages. In stage 1, the minimum freshwater obtainable in the presence of reuse and recycle opportunities was determined. In the second stage, all WMH options were examined to obtain the minimum water targets. In stage 3, the minimum number of storage tanks was determined and finally in stage 4, the minimum number of stream connections was established.

Hybrid approaches combining insight based and mathematical optimisation approaches exist. According to Oliver et al. (2008), their advantage lies in the opportunity to use targets obtained beforehand to generate alternative networks that fulfil other constraints. An additional benefit is the ability to determine the minimum cost target which was not possible with conventional insight based techniques. Foo (2010) adopted this strategy for fixed schedule problems and resource conservation networks.
Techniques relying on variable schedule

A fixed schedule oversimplifies a practical batch plant. True optimality can only be realised if the production schedule is allowed to change. As such, scheduling and wastewater minimisation should be optimised simultaneously as Majozi and Gouws (2009) observed.

Cheng and Chang (2007) developed a mathematical formulation that integrated batch production scheduling, water reuse and water treatment in a single framework. The water requirements of relevant processes were assumed proportional to the amount of material processed in a unit. The discrete time scheduling formulation of Lee et al. (2001), which was based on STN was adopted. Objectives such as freshwater and cost minimisation were investigated. Constraints associated with the network structure were also incorporated, such as the reduction of the number of storage tanks and pipe connections. The resulting model was an MINLP. No discussion regarding the solution strategy for the MINLP was provided.

Majozi (2005b) investigated the effect of both direct reuse and indirect reuse through a central storage tank for water minimisation using a flexible schedule. The scheduling framework of Majozi and Zhu (2001) that utilised the SSN and a continuous time representation was employed. Four scenarios involving single contaminants were investigated. In the first scenario, the outlet contaminant concentration was fixed at the maximum. The second scenario assumed that the flowrate through each unit was fixed, and the inlet and outlet concentrations were free to vary within defined maximum values during optimisation. The third and fourth scenarios consisted of the first and second scenarios respectively with the added option for indirect reuse via central storage. Scenario 1 resulted in a MILP, while, scenarios two, three and four resulted in MINLP problems for which global optimal solutions could not be guaranteed. Majozi and Gouws (2009) extended this formulation to account for multiple contaminants, which resulted in a MINLP. The authors employed an initialisation procedure in which a MILP reformulation of the original MINLP was solved to provide a starting point for the original MINLP. The result was a global optimum if the solution from the exact MINLP equaled that obtained from the MILP.

In another publication, Majozi (2006) proposed a formulation to determine the optimum size of a storage vessel for the maximum recovery of water, where a single contaminant was present. Gouws and Majozi (2008) extended the mathematical formulation for multiple
contaminants to include multiple storage vessels. The formulation went further to consider operations where wastewater was reused as part of the production process, thereby yielding almost zero effluent. Later, Adekola and Majozi (2011) incorporated wastewater regeneration to the formulation which further improved water reuse.

Li et al. (2010) also developed a method for the simultaneous optimisation of batch production and the batch water network with multiple contaminants. Regeneration to improve reuse opportunities was considered in addition to direct and indirect reuse opportunities. A combination of discrete and continuous time representations was employed. Specifically, the discrete feature kept track of the time intervals in which water using conditions of all operations were kept constant. The continuous time, STN scheduling framework of Ierapetritou and Floudas (1998) was adopted and a state time space superstructure was employed. The overall model was a nonconvex MINLP and was solved using a hybrid optimisation strategy integrating deterministic and stochastic components with iterations to enhance solution quality. This was computationally intensive.

Chen et al. (2011) considered the simultaneous optimisation of scheduling and batch water network in both short term and periodic operations. The formulation was based on the RTN scheduling framework of Chen and Chang (2009). The solution procedure for the resulting MINLP involved a reformulation-linearsation of the MINLP to a MILP. The MILP was solved to obtain a feasible schedule since the constraints associated with production scheduling were undisturbed during the relaxation. Next, the original MINLP was solved as an NLP by fixing the allocation binary variables obtained previously to determine a feasible water network. Although global optimality could not be guaranteed, the results were obtained in reasonable solution times.

Recently, Chaturvedhi and Bandyopadhyay (2014) developed a multiobjective formulation for the simultaneous optimisation of production scheduling and freshwater consumption. The authors developed and utilized a scheduling framework based on the STN. The main purpose of this work was to demonstrate through the use of a Pareto optimal front, trade-offs between the minimisation of freshwater and the maximisation of production, useful for selecting an appropriate schedule based on given requirements. The model was solved by optimising one objective while keeping the other fixed. The authors observed that the weighted objective
(overall cost minimisation) as adopted by many researchers was incapable of determining all the Pareto optimal points due to the nonconvexity of the model.

### 2.3.5. Optimisation of both energy and water in batch plants

Srinivas and El-Halwagi (1994) drew attention to the fact that water and energy usage in process industries can be interdependent. A large quantity of water is consumed during energy production for steam and energy is required to heat or cool water for various purposes. Grossmann et al. (2014) provided an example of such close interaction: In a heat integrated distillation column, the consumption of steam in reboilers is reduced, which in turn decreases the amount of water use in the boiler loop.

The economic objectives for heat recovery and mass exchange are generally competitive. Therefore, these two problems should be considered simultaneously to establish the complex trade-offs among raw materials, investment costs, energy and water consumption that will be necessary for lowering costs.

Literature exists for the consideration of this problem in continuous plants, the main challenge being the handling of variables that connect both systems such as temperature and mass. As a result, many of these methods have employed a sequential procedure in which water targeting is performed first followed by determining the hot and cold utilities based on the target. The techniques for solving the problem fall into two categories: conceptual or insight based and mathematical. The advantages and drawbacks are the same as for the independent investigation of energy and water. Conceptual methods are attractive because they provide some visualisation of the problem, offer some control over the solution space and provide insights for network synthesis. In addition, the independent consideration of energy and water with mathematical programming methods generated complex models that were difficult to solve. Examples of conceptual based methods in continuous plants found in literature are the two dimensional grid diagram and separate systems (Savulescu et al. 2005a,b), source demand energy composite curves (Savulescu et al., 2002; Manan et al., 2009; Alwi et al., 2011), stream merging principles (Feng et al., 2008), graphical thermodynamic rules (Sorin and Savulescu, 2004; Isafiade and Fraser, 2007, Leewongtanawit and Kim ,2009; Luo et al., 2012), etc.
Examples of mathematical programming techniques are LP methods (Bagajewicz et al., 2002; Feng et al., 2009; Sahu and Bandyopadhyay, 2012), MINLP approaches (Bogataj and Bagajewicz, 2008; Leewongtanawit and Kim, 2008; Dong et al., 2008; Kim et al., 2009; Ataei et al., 2009), etc.

Research investigating the optimisation of both energy and water in batch applications is very limited in comparison but gaining traction. Halim and Srinivasan (2011) were among the first to investigate the problem. Opportunities for water optimisation came in the form of water required for washing equipment units and heat recovery opportunities were present in the form of heating and cooling required for reactions to proceed. The authors developed a framework that integrated scheduling, direct heat integration and direct water reuse optimisation based on a sequential technique. Firstly, the process schedule was optimised to meet an economic objective. This was followed by the generation of alternate schedules through a stochastic search-based integer cut procedure. These alternate schedules were used to optimise energy and water consumption with the aim of retaining the optimality of the scheduling solution. Next, heat integration analysis and water reuse synthesis were each performed to optimise water and energy requirements on each schedule. Only direct heat integration and direct water reuse were considered in their formulation.

2.4. Research gaps

The literature review has shown that the development of robust and efficient mathematical models for batch production scheduling, which can handle a variety of problem features still remains a research focus.

Sequence dependent changeover costs were considered in scheduling formulations provided that the unique successors of tasks occurring in a unit could be identified adequately. As a result, the majority of formulations undertaking this problem focused on batch plants with sequential production paths. Regarding water minimisation, it has already been established that a true minimum for water usage in batch plants can only be obtained if production scheduling and water use optimisation are performed simultaneously. In past works, the water requirement in a unit was dictated by the task that took place in the unit only. Missing from
literature is a formulation that determines the water requirement in units based on tasks and their unique successors and minimises water usage as a result.

The optimisation of both energy and water in a plant should be investigated simultaneously due to the close interaction of energy and water systems. Missing from literature is a formulation that embeds water optimisation and energy integration within the scheduling framework straightforwardly and obtains results in reasonable computational times.

References


Clayton, R.W., 1986. Cost reductions on an edible oil refinery identified by a process integration study at Van den Berghs and Jurgens Ltd. UK. Energy Efficiency Office R&D. Energy Technology Support Unit (ETSU), Harwell Laboratory, Report No. RD/14/14.


Chapter 2. Literature Review


CHAPTER 3
A New Scheduling Formulation for Multipurpose Batch Plants with Sequence Dependent Changeover

Introduction

This chapter presents a mathematical formulation for scheduling multipurpose batch plants, where, the changeover costs associated with different sequences of tasks in units are given parameters. The formulation is based on the continuous time, SSN based framework of Seid and Majozi (2012), chosen for its proven efficiency. It is now extended to incorporate several features including: sequence dependent changeover activities, NIS storage policy, utility constraints and the storage of material in idle units. This improves the scope of problems solvable using the scheduling formulation.

3.1. Problem statement

The problem considered can be stated formally as follows.

Given the following scheduling data:

(i) production recipe including processing times of operations required to convert raw materials to product,
(ii) available units and their capacities,
(iii) availability and capacity of storage for each material,
(iv) availability of utilities,
(v) costs of raw materials,
(vi) selling price of final products,
(vii) changeover duration,
(viii) changeover costs and
(ix) time horizon of interest,
Determine the production schedule that achieves the maximum profit.

3.2. Mathematical formulation

Constraints of the original scheduling formulation of Seid and Majozi (2012) are presented for clarity. Next, new constraints and some modifications of original constraints are presented to address sequence dependent changeover. To cater for NIS and storage of material in idle units, some original constraints are modified and new constraints derived and incorporated. Finally, constraints derived to cater for utility usage are presented.

The sets, parameters and variables on which the mathematical formulation is based can be found in the nomenclature before the end of this chapter.

3.2.1. Scheduling constraints of Seid and Majozi (2012)

3.2.1.1. Allocation constraint

An equipment unit in a batch facility may be suitable to perform more than one task. Constraint (3.1) ensures that at any point in time, if a unit $j$ is active, only one task out of all the possible tasks it can perform is processed.

$$\sum_{s_{in,j}} y(s_{in,j}, p) \leq 1 \quad \forall \; j \in J, \; p \in P$$  \hspace{1cm} (3.1)

3.2.1.2. Capacity constraint

Constraint (3.2) ensures that the amount of material processed in a unit lies within the lower and upper bounds on the capacity of the unit.

$$V_{s_{in,j}}^L y(s_{in,j}, p) \leq m(s_{in,j}, p) \leq V_{s_{in,j}}^U y(s_{in,j}, p) \quad \forall \; j \in J, \; s_{in,j} \in S_{in,j}, \; p \in P$$  \hspace{1cm} (3.2)
3.2.1.3. Material balance for storage

In situations where storage is available for materials, Constraint (3.3) states that the amount of material stored at a specific point in time consists of the amount of material stored at the previous time point, adjusted for the difference between the amount of material produced by tasks at the previous time point, \( p - 1 \) and the amount of material consumed by tasks at the current time point \( p \). Constraint (3.4) is the storage constraint of a final product, no consumption takes place.

\[
q_s(s, p) = q_s(s, p - 1) - \sum_{s_{u,j} \in S_{w,j}} \rho_{s_{w,j}} m_u(s_{in,j}, p) + \sum_{s_{u,j} \in S_{w,j}} \rho_{s_{w,j}} m_u(s_{in,j}, p - 1) \\
\forall s \in S, \ p \in P, \ p > p_1 \tag{3.3}
\]

\[
q_s(s^p, p) = q_s(s^p, p - 1) + \sum_{s_{u,j} \in S_{w,j}} \rho_{s_{w,j}} m_u(s_{in,j}, p) \quad \forall s^p \in S^p, \ p \in P \tag{3.4}
\]

3.2.1.4. Duration constraint: Duration as a function of batch size

Constraint (3.5) describes the duration of a task as a linear function of the associated batch size.

\[
t_p(s_{in,j}, p) \geq t_u(s_{in,j}, p) + \alpha(s_{in,j}) + \beta(s_{in,j}) + m_u(s_{in,j}, p) \]

\[
\forall j \in J, \ s_{in,j} \in S_{in,j}, \ p \in P \tag{3.5}
\]

3.2.1.5. Sequence constraints

These constraints cater for the timings of tasks to avoid overlap and ensure feasibility, that is, the starting time of a new task should take place after the finishing time of a previous task.

Same task in the same unit

For different batches of the same task being processed in the same unit, the time at which a new batch begins processing must be after the previous batch is completed. This requirement is captured by Constraint (3.6).
Chapter 3. A New Scheduling Formulation for Multipurpose Batch Plants with Sequence Dependent Changeover

\[ t_u(s_{in,j}, p) \geq t_p(s_{in,j}, p-1) \quad \forall \ j \in J, s_{in,j} \in S^*_{in,j}, p \in P, p > p_1 \]  \hfill (3.6)

**Different tasks in the same unit**

Similarly, Constraint (3.7) is for a unit that can perform many tasks. It states that the start time of a new task should be after the finish time of any task that took place at a previous time.

\[ t_u(s_{in,j}, p) \geq t_p(s'_{in,j}, p-1) \]

\[ \forall \ j \in J, s_{in,j} \neq s'_{in,j}, s_{in,j}, s'_{in,j} \in S^*_{in,j}, p \in P, p > p_1 \]  \hfill (3.7)

**Different tasks in different units if an intermediate is produced from one unit**

These constraints deal with the appropriate sequencing of tasks where an intermediate is concerned, that is, for different tasks that produce and consume the same state. The starting time of a consuming task at time point \( p \) must be after the finishing time of any producing task at \( p - 1 \).

Constraints (3.8) and (3.9) work together. In both constraints, if an intermediate is produced from a task performed in unit \( j \) at time point \( p - 1 \) and consumed by another task performed in unit \( j' \) at time point \( p \), the binary variable \( t(j, p) \) assumes a value of 1. Constraint (3.8) states that if this is the situation, the amount of intermediate produced may exceed the storage available, \( q_i(s, p) \). Constraint (3.9) ensures that the time at which the task consuming the intermediate occurs after the completion of the task producing the intermediate. If on the other hand, the intermediate is not consumed immediately by the task performed in \( j' \) at time point \( p \), the binary variable \( t(j, p) \) assumes a value of 0 and Constraint (3.9) is relaxed while in Constraint (3.8), the amount of intermediate produced should not exceed the storage available.

© University of Pretoria
Chapter 3. A New Scheduling Formulation for Multipurpose Batch Plants with Sequence Dependent Changeover

\[ \rho^p_{s_{m,j}} m_u(s_{m,j}, p-1) \leq q(s, p) + V^U_{j,p}(j, p) \]
\[ \forall j \in J, s_{m,j} \in S^p_{m,j}, p \in P, p > p_1 \]  
\begin{equation} \tag{3.8} \end{equation}

\[ t_{s}(s_{m,j}^{i}, p) \geq t_{p}(s_{m,j}, p-1) - H(2 - \gamma(s_{m,j}, p-1) - t(j, p)) \]
\[ \forall j \in J, s_{m,j} \in S^p_{m,j}, s_{m,j}^{i} \in S^{sc}_{m,j}, p \in P, p > p_1 \]  
\begin{equation} \tag{3.9} \end{equation}

Different tasks in different unit if an intermediate is produced from more than one unit

Constraint (3.10) caters for the situation where an intermediate state is produced from more than one unit (or from just one unit). It states that the amount of intermediate consumed at time point \( p \) may come from either storage or from other producing tasks in different units, that is \( t(j, p) \) is 1. If this is the case, Constraint (3.9) is also activated. If \( t(j, p) \) is 0 however, Constraint (3.10) is relaxed as well.

Constraint (3.10) although nonlinear, can be linearised exactly using Glover transformations (Glover, 1975).

\[ \sum_{s_{m,j} \in S^p_{m,j}} \rho^p_{s_{m,j}} m_u(s_{m,j}, p-1) \leq q(s, p-1) + \sum_{s_{m,j} \in S^p_{m,j}} \rho^p_{s_{m,j}} m_u(s_{m,j}, p-1) f(j, p) \]
\[ \forall j \in J, p \in P, p > p_1 \]  
\begin{equation} \tag{3.10} \end{equation}

3.2.1.6. Usage of previously stored states

Constraint (3.11) ensures that if a task occurring at time point \( p \) consumes intermediate material which was previously stored at \( p-1 \) but was produced by a task at time point \( p-2 \), the time at which it begins is after the completion of the task at \( p-2 \).

\[ t_{s}(s_{m,j}^{i}, p) \geq t_{p}(s_{m,j}, p-1) - H(1 - \gamma(s_{m,j}, p-2)) \]
\[ \forall j \in J, s_{m,j} \in S^p_{m,j}, s_{m,j}^{i} \in S^{sc}_{m,j}, p \in P, p > p_2 \]  
\begin{equation} \tag{3.11} \end{equation}
3.2.1.7. Constraints for FIS policy

Constraint (3.12) states that any intermediate state can be stored in a storage vessel provided that the capacity is not exceeded. Otherwise, the intermediate state should be consumed immediately after it is produced or not at all. A new binary variable \( x(s, p) \) is defined to signify the presence or absence of storage capacity. In the absence of storage capacity, \( x(s, p) = 0 \), the amount of intermediate produced is limited by the capacity of the consuming unit.

Constraint (3.13) caters for the sequencing of tasks whether storage is available or not. From constraint (3.9), the starting time of a task in unit \( j' \) that consumes an intermediate state \( s \) at time point \( p \) must equal the finishing time of a task performed in unit \( j \) that produced the intermediate at the previous time point \( p - 1 \). Constraint (3.13) enforces this provided that there is no storage available for storing state \( s \), \( x(s, p) = 0 \), and that the consuming task and producing task are active at time point \( p \) and \( p - 1 \) respectively. In a situation where storage is available to store state \( s \), \( x(s, p) = 1 \), the starting time of a task in unit \( j' \) that consumes state \( s \) at time point \( p \) may not necessarily equal the finishing time of a task in unit \( j \) that produced the state previously. In this situation, the constraint is relaxed. Constraint (3.13) ensures that the use of a storage vessel is feasible both in terms of capacity and real time and not just with respect to time points alone.

\[
\sum_{s_{in,j} \in S^{\text{ip}}_{\text{in,j}}} \rho_{s_{in,j}} m_{u}(s_{in,j}, p-1) + q_{u}(s, p-1) \leq QS^U + \sum_{j \in J} V_{j}^U (1 - x(s, p))
\]

\( \forall j \in J, s \in S, p \in P, p > p_i \)  \hspace{1cm} (3.12)

\[
t_u(s_{in,j}, p) \leq t_u(s_{in,j}, p-1) + H(2 - y(s_{in,j}, p) - y(s_{in,j}, p-1)) + H(x(s, p))
\]

\( \forall j \in J, s_{in,j} \in S^{\text{ip}}_{\text{in,j}}, s_{in,j} \in S^{\text{sc}}_{\text{in,j}}, s \in S, p \in P, p > p_i \)  \hspace{1cm} (3.13)
3.2.1.8. Storage constraints when an idle unit stores material produced previously

Units in which producing tasks take place can store the intermediate produced if the units are idle. Constraint (3.14) states that the amount of state $s$ stored at any time point must not exceed the maximum capacity of the storage. Storage available here is defined as the sum of the capacity of intermediate storage and the amount of intermediate produced by the various tasks at time point $p$.

Constraint (3.15) states that state $s$ produced at $p-1$ can be stored for a while in the unit that produced it at consecutive time points.

Constraint (3.16) ensures that if a state is stored in a unit at a time point, it cannot perform a task at that same time point.

\[
q_s(s, p) \leq QS^U + \sum_{s_{in}, s_{out}} u(s_{in}, j, p) \quad \forall j \in J, p \in P, s \in S
\]  

(3.14)

\[
u(s_{in}, j, p) \leq \rho_{s_{in}, j} m_s(s_{in}, j, p-1) + u(s_{in}, j, p-1)
\quad \forall j \in J, s_{in}, j \in S_{in, j}^{up}, p \in P, p > p_1
\]  

(3.15)

\[
u(s_{in}, j, p) \leq V^U_j - \left(1 - \sum_{s_{in}, j \in S_{in, j}^{up}} y(s_{in}, j, p)\right) \quad \forall j \in J, p \in P, s_{in}, j \in S_{in, j}^{up}
\]  

(3.16)

3.2.1.9. Time horizon constraints

Constraints (3.17) and (3.18) ensure that the time at which states are used and produced occurs within the time horizon of interest.

\[t_u(s_{in}, j, p) \leq H \quad \forall j \in J, p \in P, s_{in}, j \in S_{in, j}
\]  

(3.17)

\[t_b(s_{in}, j, p) \leq H \quad \forall j \in J, p \in P, s_{in}, j \in S_{in, j}
\]  

(3.18)
3.2.2. Changeover constraints

The changeover binary variable, \( x_{ch}(s_{m,j}, s'_{m,j}, p) \) is equal to 1 if task \( s_{m,j} \) is processed before task \( s'_{m,j} \) in unit \( j \) at time point \( p \) or zero otherwise. Constraints (3.19)-(3.21) provide a relationship between the assignment binary variable \( y(s_{m,j}, p) \) of both tasks in the same unit and the changeover binary variable. Constraint (3.19) indicates that for changeover to occur, the processing task \( s_{m,j} \) must take place. However, the fact that task \( s_{m,j} \) takes place in unit \( j \) does not necessarily imply changeover. Constraint (3.20) is similar and states that for changeover to occur, the succeeding task, \( s'_{m,j} \) must take place at a later time point. Constraints (3.20) and (3.21) together, account for any intermediate tasks that could be processed after processing \( s_{m,j} \) but before processing \( s'_{m,j} \). If such intermediate tasks occur, the changeover binary variable, \( x_{ch}(s_{m,j}, s'_{m,j}, p) \) is zero. Constraint (3.22) captures the condition that at any given time point, at most one task can be the immediate successor to the current task in a unit. Constraints (3.19) and (3.21) are similar to constraints in Kabra et al. (2013).

\[
x_{ch}(s_{m,j}, s'_{m,j}, p) \leq y(s_{m,j}, p) \quad \forall \ j \in J, \ s_{m,j}, s'_{m,j} \in S_{m,j}, \ p \in P, \ p < P \quad (3.19)
\]

\[
x_{ch}(s_{m,j}, s'_{m,j}, p') \leq y(s'_{m,j}, p) + \sum_{p < p' < p} \sum_{s_{m,j} \in S_{m,j}} y(s_{m,j}, p')
\quad \forall \ j \in J, \ s_{m,j}, s'_{m,j} \in S_{m,j}, \ p', p \in P, \ p' < p, \ p' < P \quad (3.20)
\]

\[
x_{ch}(s_{m,j}, s'_{m,j}, p') \geq y(s_{m,j}, p') + y(s'_{m,j}, p) - 1 - \sum_{p < p' < p} \sum_{s_{m,j} \in S_{m,j}} y(s_{m,j}, p')
\quad \forall \ j \in J, \ s_{m,j}, s'_{m,j} \in S_{m,j}, \ p', p \in P, \ p' < p, \ p' < P \quad (3.21)
\]

\[
\sum_{s'_{m,j} \in S_{m,j}} x_{ch}(s_{m,j}, s'_{m,j}, p) \leq 1 \quad \forall \ j \in J, \ s_{m,j} \in S_{m,j}, \ p \in P, \ p < P \quad (3.22)
\]

Constraint (3.5) determines the finish time of a production task in a unit. Constraint (3.23) defines the finish time of the changeover operation as the sum of the finish time of the production task and the changeover duration, which depends on the succeeding production
task. Constraint (3.24) ensures that the start time of the succeeding production task is after the completion of the changeover operation. If no changeover operation takes place, \( x_{ch}(s_{m,j}, s'_{m,j}, p) \) is equal to 0 and the start time of the succeeding task is simply after the finish time of production of the previous task.

\[
t_{w_{out}}(s_{m,j}, p) = t_p(s_{m,j}, p) + \sum_{s_{m,j}} \left( t_{ch}(s_{m,j}, s'_{m,j}) \cdot x_{ch}(s_{m,j}, s'_{m,j}, p) \right)
\]

\[\forall j \in J, \ s_{m,j} \in S_{m,j}, \ p \in P \quad (3.23)\]

\[
t_{u}(s'_{m,j}, p) \geq t_{w_{out}}(s_{m,j}, p') - H(1 - x_{ch}(s_{m,j}, s'_{m,j}, p'))
\]

\[\forall j \in J, \ s_{m,j}, s'_{m,j} \in S_{m,j}, \ p, p' \in P, \ p' < p, \ p' < P \quad (3.24)\]

### 3.2.3. New storage constraints: NIS and storage in an idle unit

The proposed formulation caters for states with no intermediate storage and the possibility to store material in idle processing units. The original scheduling framework of Seid and Majozi (2012) allows for an intermediate state to be stored in the unit that produced it, if the unit remains idle after the producing task has ended. The formulation did not consider storage of any intermediate state in any available idle unit. The necessary modifications and additions are described here.

Constraint (3.1) is modified to (3.25) and stipulates that if a unit is active at any time point, the unit is either processing a task or storing a state. The binary variable, \( x_u(s, j, p) \) is 1 if state \( s \) is stored in unit \( j \) at time point \( p \). The constraint also ensures that only one state can be stored in a unit at a given time point, if the unit is storing material.

\[
\sum_{s \in S^u} x_u(s, j, p) + \sum_{s_{m,j} \in S_{m,j}} y(s_{m,j}, p) \leq 1 \quad \forall j \in J^u, \ p \in P
\]

(3.25)

The storage constraints for an intermediate state in a storage vessel are modified to accommodate the possibility of storage of the intermediate state in idle units. Constraint (3.3) is modified to Constraint (3.26). The parameter \( \delta_{s_{m,j}}^{ip} \) determines the fraction of intermediate state produced by a task that is sent to the intermediate storage vessel. Similarly, the
Chapter 3. A New Scheduling Formulation for Multipurpose Batch Plants with Sequence Dependent Changeover

The parameter $\delta^{sc}_{s_{in,j}}$ determines the fraction of intermediate state consumed by a task that was obtained from the intermediate storage vessel. The values of these parameters are obtained after optimisation.

$$q_s(s, p) = q_s(s, p-1) - \sum_{s_{in,j} \in S^{sp}_{s_{in,j}}} \delta^{sc}_{s_{in,j}} m_u(s_{in,j}, p) + \sum_{s_{in,j} \in S^{sp}_{s_{in,j}}} \delta^{sp}_{s_{in,j}} m_u(s_{in,j}, p-1)$$

$$\forall \ p \in P, \ s \in S^{st} \quad (3.26)$$

For states with NIS, Constraint (3.8) is modified to Constraint (3.27). The modified constraint states that the amount of intermediate produced is limited by the total capacity of idle units available for storage, if it is not consumed immediately. For other states, Constraint (3.8) is modified to Constraint (3.28) to include the possibility of storing material in idle units.

$$\rho^{sp}_{s_{in,j}} m_u(s_{in,j}, p-1) \leq \sum_{j' \in J^s} V^{U}_{j'} x_u(s, j', p) + V^{U}_{j'} t(j, p)$$

$$\forall \ j \in J, \ p \in P, \ s_{in,j} \in S^{sp}_{s_{in,j}}, \ s \in S^{nis} \quad (3.27)$$

$$\rho^{sp}_{s_{in,j}} m_u(s_{in,j}, p-1) \leq q_s(s, p) + \sum_{j' \in J^s} V^{U}_{j'} x_u(s, j', p) + V^{U}_{j'} t(j, p)$$

$$\forall \ j \in J, \ p \in P, \ s_{in,j} \in S^{sp}_{s_{in,j}}, \ s \in S^{st} \quad (3.28)$$

For states with NIS, Constraint (3.10) is modified to Constraint (3.29). The constraint states that the amount of intermediate consumed at $p$ could be obtained from tasks producing the intermediate at $p-1$, if the intermediate is used immediately $t(j, p) = 1$, or from stored intermediate in idle units. If $t(j, p)$ is zero, the amount of intermediate consumed at $p$ must be provided from stored material in idle units.

$$\sum_{s_{in,j} \in S^{sp}_{s_{in,j}}} \rho^{sc}_{s_{in,j}} m_u(s_{in,j}, p) \leq \sum_{s_{in,j} \in S^{sp}_{s_{in,j}}} \rho^{sp}_{s_{in,j}} m_u(s_{in,j}, p-1) t(j, p) + \sum_{j' \in J^s} q_{j'}(s, j', p-1)$$

$$\forall \ j \in J, \ p \in P, \ s \in S^{nis} \quad (3.29)$$

For other states, Constraint (3.10) is modified to Constraint (3.30) to account for the possibility of storing material in idle units. The amount of intermediate consumed at $p$ could...
be obtained from stored material in the intermediate storage vessel, from tasks producing the intermediate at \( p - 1 \), if the intermediate is used immediately or from stored intermediate in idle units.

\[
\sum_{s_{m,j} \in S_{m,j}} \rho_{s_{m,j}}^{sc} m_u(s_{m,j}, p) \leq q_u(s, p - 1) + \sum_{s_{m,j} \in S_{m,j}} \rho_{s_{m,j}}^{sp} m_u(s_{m,j}, p - 1) q_j(p, j) + \sum_{j \in J^p} q_u(s, j', p - 1) \quad \forall j \in J, p \in P, s \in S^u
\]

Constraints (3.29) and (3.30) are linearised using Glover (1975) transformations.

Constraint (3.12) is modified to Constraint (3.31) for states with NIS and to Constraint (3.32) for states with FIS to account for material storage in idle units. The constraints state that the amount of intermediate produced is either consumed by tasks or stored in available storage units.

\[
\sum_{s_{m,j} \in S_{m,j}} \rho_{s_{m,j}}^{sp} m_u(s_{m,j}, p - 1) \leq \sum_{s_{m,j} \in S_{m,j}} \rho_{s_{m,j}}^{sc} m_u(s_{m,j}, p) + \sum_{j \in J^p} q_u(s, j', p) \quad \forall j \in J, p \in P, s \in S^{nis}
\]

\[
\sum_{s_{m,j} \in S_{m,j}} \rho_{s_{m,j}}^{sp} m_u(s_{m,j}, p - 1) + q_u(s, p - 1) \leq QU + \sum_{j \in J^p} V_j(1 - x(s, j)) + \sum_{j \in J^p} q_u(s, j', p) \quad \forall j \in J, p \in P, s \in S^a
\]

Constraint (3.13) is modified to (3.33) for states with NIS. Constraints (3.33) and (3.9) together ensure that the starting time of a task consuming a state and the finishing time of a task producing the state coincide if the state is used immediately, that is, \( t(j, p) = 1 \).

\[
t_u(s_{m,j}, p - 1) \leq t_p(s_{m,j}, p - 1) + H(2 - y(s_{m,j}, p) - y(s_{m,j}, p - 1)) \\
\quad \forall j, j' \in J, p \in P, s_{m,j} \in S^{sp}_{m,j}, s_{m,j} \in S^{sc}_{m,j}, s \in S^{nis}
\]

The following constraints account for the storage of material in idle processing units. Constraint (3.34) states that the amount of intermediate stored in an idle unit consists of the
amount of intermediate stored in the unit previously and the difference between the amount of intermediate produced at a previous time point that is stored in the idle unit and the amount of stored intermediate in the idle unit that is consumed by tasks at the present time point. The constraint includes the possibility to store the intermediate produced by a task in more than one idle unit (the unit that produced it included). The parameter, \( \gamma_{s_{m,j}} \), determines the fraction of intermediate produced by task \( s_{m,j} \) that is sent to the idle unit \( j \) for storage. Similarly, \( \gamma_{s_{m,j}} \) determines the fraction of stored intermediate in idle unit \( j \) that is consumed by task \( s_{m,j} \). The values of these parameters are obtained after optimisation. The amount of material stored in all processing units at the beginning of the time horizon is zero.

\[
q_u(s,j,p) = q_u(s,j,p-1) + \sum_{s_{m,j} \in S_{s_{m,j}}^{s_{m,j}}} \gamma_{s_{m,j}} m_u(s_{m,j},j,p-1) - \sum_{s_{m,j} \in S_{s_{m,j}}^{s_{m,j}}} \gamma_{s_{m,j}} m_u(s_{m,j},j,p)
\]

\[
\forall j \in J_{st}, j' \in J, p \in P, s \in (S_{st} \cup S_{st}') \quad (3.34)
\]

Constraints (3.35) and (3.36) ensure that if the intermediate produced by a task is sent to more than one storage unit, the sum of the fractions is equal to the amount of intermediate produced. Constraint (3.35) is for states with intermediate storage where storage consists of idle processing units and the intermediate storage vessel. Constraint (3.36) is for states with NIS, where storage consists of idle processing units.

\[
\sum_{j \in J_{st}} \gamma_{s_{m,j}} m_u(s_{m,j},j',p) + \delta_{s_{m,j}} m_u(s_{m,j},p) = \rho_{s_{m,j}} m_u(s_{m,j},p)
\]

\[
\forall j \in J, p \in P, s_{m,j} \in S_{st}^{s_{m,j}}, s \in S_{st} \quad (3.35)
\]

\[
\sum_{j \in J_{st}} \gamma_{s_{m,j}} m_u(s_{m,j},j',p) = \rho_{s_{m,j}} m_u(s_{m,j},p)
\]

\[
\forall j \in J, p \in P, s_{m,j} \in S_{st}^{s_{m,j}}, s \in S_{nis} \quad (3.36)
\]

Constraints (3.37) and (3.38) are similar and ensure that if the intermediate consumed by a task is obtained from more than one storage unit, the sum of the fractions is equal to the amount of intermediate consumed. Constraints (3.37) and (3.38) are for states with intermediate storage and NIS states respectively.
Chapter 3. A New Scheduling Formulation for Multipurpose Batch Plants with Sequence Dependent Changeover

\[
\sum_{j \in J^p} \gamma_{s_m,j}^{sc} m_u(s_{m,j}, j', p) + \delta_{s_m,j}^{sc} m_u(s_{m,j}, p) = \rho_{s_m,j}^{sc} m_u(s_{m,j}, p)
\]

\(\forall j \in J, p \in P, s_{m,j} \in S_{in,j}^{sc}, s \in S^{st}\)

(3.37)

\[
\sum_{j \in J^p} \gamma_{s_m,j}^{sc} m_u(s_{m,j}, j', p) = \rho_{s_m,j}^{sc} m_u(s_{m,j}, p)
\]

\(\forall j \in J, p \in P, s_{m,j} \in S_{in,j}^{sc}, s \in S^{mis}\)

(3.38)

Constraint (3.39) ensures that the amount of intermediate stored in an idle unit after a task has occurred is limited by the capacity of the idle unit. Likewise, Constraint (3.40) ensures that the amount of stored intermediate from an idle unit that is consumed by a task is limited by the capacity of the unit in which the task occurs.

\[
\gamma_{s_m,j}^{up} m_u(s_{m,j}, j', p) \leq V^{ui}_j x_u(s, j', p)
\]

\(\forall j \in J, j' \in J^{st}, p \in P, s_{m,j} \in S_{in,j}^{up}, s \in (S^{mis} \cup S^{st})\)

(3.39)

\[
\gamma_{s_m,j}^{st} m_u(s_{m,j}, j', p) \leq V^{ui}_j x_u(s, j', p)
\]

\(\forall j \in J, j' \in J^{st}, p \in P, s_{m,j} \in S_{in,j}^{sc}, s \in (S^{mis} \cup S^{st})\)

(3.40)

Constraints (3.41) and (3.42) provide relationships between the amount of material stored in a unit and \(x_u(s, j, p)\). Constraint (3.41) states that the amount of intermediate stored in an idle unit is limited by the capacity of the idle unit. Constraint (3.42) ensures that if intermediate is stored in an idle unit, it is greater than a specified minimum.

\[
q_u(s, j, p) \leq V^{ui}_j x_u(s, j, p)
\]

\(\forall j \in J^{st}, s \in S^{st}, p \in P\)

(3.41)

\[
q_u(s, j, p) \geq m \cdot x_u(s, j, p)
\]

\(\forall j \in J^{st}, s \in S^{st}, p \in P\)

(3.42)

3.2.4. Utility constraints

Constraint (3.43) calculates the amount of utility required for processing a task, where, \(f(s_{m,j})\) and \(g(s_{m,j})\) are the constant and variable terms of the amount of utility consumed.
Constraint (3.44) states that the total amount of utility consumed at any time point is limited by an upper bound.

\[
Q(s_{in,j}, u, p) = \sum_{s_{in,j} \in S_{in,j}} \left[ f(s_{in,j}) \cdot y(s_{in,j}, p) + g(s_{in,j}) \cdot m_u(s_{in,j}, p) \right]
\forall \ u \in U, \ j \in J, \ p \in P, \ s_{in,j} \in S_{in,j}
\tag{3.43}
\]

\[
\sum_{s_{in,j} \in S_{in,j}} Q(s_{in,j}, u, p) \leq Q^U(u) \quad \forall \ u \in U, \ j \in J, \ p \in P
\tag{3.44}
\]

Constraint (3.45) states that the level of utility at each time point consists of the level of utility available at the previous time point, increased by utility made available due to tasks that used the utility ending and reduced by utility consumed due to tasks that require utility beginning. Constraint (3.46) defines the level of utility at the first time point.

\[
L(u, p) = L(u, p - 1) - \sum_{s_{in,j} \in S_{in,j}} Q(s_{in,j}, u, p) + \sum_{s_{in,j} \in S_{in,j}} Q(s_{in,j}, u, p - 1)
\forall \ u \in U, \ j \in J, \ p \in P
\tag{3.45}
\]

\[
L(u, p) = Q^U(u) - \sum_{s_{in,j} \in S_{in,j}} Q(s_{in,j}, u, p) \quad \forall \ u \in U, \ j \in J
\tag{3.46}
\]

Constraint (3.45) monitors the level of utility with respect to time points. It is possible that an overlap in real time can occur between a task using utility \( u \) that ends at time point \( p - 1 \) and a task in another unit beginning at time point \( p \) that also requires utility \( u \). A consequence of this is that the total utility requirement of these tasks may exceed the level of utility available, leading to infeasible solutions. This is as a result of unit specific time points, where tasks taking place in different units at the same time point can take place at different real times. Constraints (3.47)-(3.49) address this. Constraint (3.47) states that the amount of utility required to perform a task at the present time point is less than the level of utility available at the previous time point and the amount of utility made available by the completion, at the previous time point of any tasks using the utility in other units. The binary variable, \( z(j, j', p) \) is 1 if the amount of utility made available by the completion of tasks in unit \( j' \) is used immediately by the present task in unit \( j \). Constraint (3.48) ensures that if the amount of utility made available by a task which occurred at \( p - 1 \) is used by a task occurring at \( p \), the
start time of the task at $p$ must be after the finishing time of the task at $p−1$. Constraint (3.49) ensures that for two tasks which use the same utility taking place in different units, the starting time of the task at the present time point must be later than the finishing time of the task at a previous time point.

$$Q(s_{in,j}, u, p) \leq L(u, p−1) + \sum_{s_{in,j} \in S_{in,j}} Q(s_{in,j}, u, p−1) \cdot z(j, j′, p)$$
\[\forall \ u \in U, \ j, j′ \in J, \ j \neq j′, \ p \in P, \ s_{in,j} \in S_{in,j}^{u} \tag{3.47}\]

$$t_a(s_{in,j}, p) \geq t_p(s_{in,j}, p−1) - H(2 - y(s_{in,j}, p−1) - z(j, j′, p))$$
\[\forall \ j, j′ \in J, \ j \neq j′, \ p \in P, \ s_{in,j}, s_{in,j′} \in S_{in,j}^{u} \tag{3.48}\]

$$t_a(s_{in,j}, p) \geq t_p(s_{in,j′, p′})$$
\[\forall \ j, j′ \in J, \ j \neq j′, \ p, p′ \in P, \ p > p′, \ s_{in,j}, s_{in,j′} \in S_{in,j}^{u} \tag{3.49}\]

### 3.2.5. Objective function

The objective is to maximise profit while taking into account production costs and changeover costs. Equation (3.50) calculates the revenue from product, Equation (3.51) calculates the cost of raw material and Equation (3.52) calculates the cost associated with sequence dependent changeover. Equation (3.53) is the objective to be maximised.

Revenue = \[\sum_{s} q_s(s, p)CP(s) \quad \forall \ p = P \tag{3.50}\]

Costraw = \[\sum_{s_{in,j}^{u} \in S_{in,j}} \sum_{p \in P} CR(s)\rho_{s_{in,j}}^{u} m_a(s_{in,j}, p) \tag{3.51}\]

CostChangeover = \[\sum_{p \in P} \sum_{s_{in,j}^{u} \in S_{in,j}} \sum_{s_{in,j}′} CCH(s_{in,j}, s_{in,j}′) v_{ch}(s_{in,j}, s_{in,j}′, p) \tag{3.52}\]

Objective = Revenue – Costraw – CostChangeover \tag{3.53}
The resulting formulation is a mixed integer linear problem, MILP for which a global optimum solution is guaranteed.

3.3. Case study

This case study originally from Maravelias and Grossmann (2003) was adapted by Gimenez et al. (2009) to consider both sequence dependent and independent changeovers when two different tasks take place in the same processing unit. In the multipurpose batch facility, two products, P1 and P2 are produced from two raw materials RM1 and RM2 through four reactions, R1, R2, R3 and R4. Three reactors, R-101, R-102 and R-103 are available to perform selected reactions. Reactors R-101 and R-102 are suitable to perform reactions R1 and R2, while R-103 is suitable to perform reactions R3 and R4. The STN and SSN representations of the facility are illustrated in Figure 3.1.

Figure 3.1. STN (a) and SSN (b) representations of case study
Table 3.1 provides information regarding the production process. Intermediate, INT2 has been assigned no intermediate storage (NIS). Reactions R1 and R3 require heating, while reactions R2 and R4 require cooling. Data pertaining to the utility requirements of the tasks are given in Table 3.2. HS is hot steam, CW is cooling water.

### Table 3.1. Production data

<table>
<thead>
<tr>
<th>Task</th>
<th>Unit</th>
<th>Min batch size (kg)</th>
<th>Max batch size (kg)</th>
<th>$\alpha$ (h)</th>
<th>$\beta$ (h/kg)</th>
<th>Material state</th>
<th>Initial inventory (kg)</th>
<th>Max storage (kg)</th>
<th>Revenue/cost ($/kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>R-101</td>
<td>40</td>
<td>80</td>
<td>0.5</td>
<td>0.025</td>
<td>RM1 (s1)</td>
<td>400</td>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>R-102</td>
<td>25</td>
<td>50</td>
<td>0.5</td>
<td>0.04</td>
<td>RM2 (s2)</td>
<td>400</td>
<td>1000</td>
<td>15</td>
</tr>
<tr>
<td>R2</td>
<td>R-101</td>
<td>40</td>
<td>80</td>
<td>0.75</td>
<td>0.0375</td>
<td>INT1 (s3)</td>
<td>0</td>
<td>50</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>R-102</td>
<td>25</td>
<td>50</td>
<td>0.75</td>
<td>0.06</td>
<td>INT2 (s4)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R3</td>
<td>R-103</td>
<td>40</td>
<td>80</td>
<td>0.25</td>
<td>0.0125</td>
<td>INT3 (s5)</td>
<td>0</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>R4</td>
<td>R-103</td>
<td>40</td>
<td>80</td>
<td>0.5</td>
<td>0.025</td>
<td>P1 (s6)</td>
<td>0</td>
<td>1000</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P2 (s7)</td>
<td>0</td>
<td>1000</td>
<td>40</td>
</tr>
</tbody>
</table>

### Table 3.2. Utility requirements

<table>
<thead>
<tr>
<th>Task</th>
<th>Unit</th>
<th>Utility</th>
<th>$f$ (kg/min)</th>
<th>$g$ (kg/min kg)</th>
<th>Max utility (kg/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>R-101</td>
<td>HS</td>
<td>6</td>
<td>0.25</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>R-102</td>
<td>HS</td>
<td>4</td>
<td>0.25</td>
<td>30</td>
</tr>
<tr>
<td>R2</td>
<td>R-101</td>
<td>CW</td>
<td>4</td>
<td>0.3</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>R-102</td>
<td>CW</td>
<td>3</td>
<td>0.3</td>
<td>30</td>
</tr>
<tr>
<td>R3</td>
<td>R-103</td>
<td>HS</td>
<td>8</td>
<td>0.2</td>
<td>30</td>
</tr>
<tr>
<td>R4</td>
<td>R-103</td>
<td>CW</td>
<td>4</td>
<td>0.5</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 3.3 provides changeover information which is characterised by cleaning tasks. The relevant task sequences and corresponding cleaning times and costs are provided. Two cleaning in place units C-101 and C-102 are available for the cleaning operation. The objective in this case study was to determine the maximum profit over a time horizon of 8 h.

### Table 3.3. Cleaning data

<table>
<thead>
<tr>
<th>Task Sequence</th>
<th>Unit</th>
<th>Cleaning unit</th>
<th>Cleaning time (h)</th>
<th>Cleaning cost($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1-R2</td>
<td>R-101</td>
<td>C-101</td>
<td>0.15</td>
<td>80</td>
</tr>
<tr>
<td>R2-R1</td>
<td>R-101</td>
<td>C-101</td>
<td>0.18</td>
<td>100</td>
</tr>
<tr>
<td>R1-R2</td>
<td>R-102</td>
<td>C-101</td>
<td>0.10</td>
<td>45</td>
</tr>
<tr>
<td>R2-R1</td>
<td>R-102</td>
<td>C-101</td>
<td>0.12</td>
<td>60</td>
</tr>
<tr>
<td>R3-R4</td>
<td>R-103</td>
<td>C-102</td>
<td>0.15</td>
<td>80</td>
</tr>
<tr>
<td>R4-R3</td>
<td>R-103</td>
<td>C-102</td>
<td>0.15</td>
<td>80</td>
</tr>
</tbody>
</table>
Chapter 3. A New Scheduling Formulation for Multipurpose Batch Plants with Sequence Dependent Changeover

The model for the case study consists of Constraints (3.2), (3.4)-(3.7), (3.9), (3.11) and (3.17)-(3.53). The model was solved using GAMS 22.0 software on a machine with Intel(R) Core(TM) i7-2670QM, 2.2 GHz processor and 4.0GB RAM and CPLEX as the MILP solver.

3.3.1. Results and Discussion

The value of the objective function, maximisation of profit is $3093.04. The resulting production schedule illustrated in Figure 3.2 is equivalent to that obtained by Gimenez et al. (2009). No improvement on this objective is possible as will be explained.

![Figure 3.2. Resulting production schedule for case study](image)

From Figure 3.1, it can be noted that reactions R1, R3 and R4 can only occur after R2 has taken place. INT2, a product of reaction R2, has been assigned no intermediate storage. As such, INT2 is used immediately it is produced or stored in an idle unit. In Figure 3.2, R2 is produced at $p1$ in unit R-101 and R-102 but cannot be used immediately, as R4 requires INT3, yet unavailable to be performed. The finish time of R2 in R-101 is 2.37 h while the finish time of R2 in R-102 is 2.25 h. As a result, 17.5 kg of INT2 available at 2.25 h is sent to R-103, the only other available unit, to allow R-102 perform R1. At 2.37 h, the 33.64 kg of INT2 produced in R-101 remained in R-101, the only available unit at the time. Unit R-101 is unavailable for processing any tasks while it is storing INT2. If R-102 also stored INT2 it produced, no unit would have been available to perform reaction R1 which was necessary for continuity of the process. Once R1 in R-102 produced INT1 at 3.87 h, R4 occurred in R-103 using the17.5 kg of INT2 stored in R-103 and 6.5kg of INT2 stored in R-101. At 5.37 h, R4
occurs again using the remaining 27.14 kg of INT2 stored in R-101. Since intermediate storage exists for all other states, any amount produced that was not used immediately was stored. The only difference between this schedule and that obtained by Gimenez et al. (2009) is that 5.51 kg of INT1 produced in R-102 at 5.37 h is stored in intermediate storage, while in Gimenez et al. (2009) it was stored in R-101. This is a trivial difference since the cost of storage was not included in the objective. Changeover occurred at 2.25 h in R-102 to prepare for reaction R1 and occurred at 7 h to prepare R-103 for reaction R3.

The computational statistics for this case study are presented in Table 3.4. The statistics of model GHM are those reported by Gimenez et al. (2009) for the case study. The proposed model is computationally superior compared to model GHM.

Table 3.4. Computational statistics of case study

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>CPU time (s)</th>
<th>Nodes</th>
<th>RMILP ($)</th>
<th>MILP ($)</th>
<th>B.V.</th>
<th>C.V.</th>
<th>Constraints</th>
<th>Non-zeros</th>
<th>Relative gap %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>6</td>
<td>1.5</td>
<td>755</td>
<td>5550.5</td>
<td>3093</td>
<td>226</td>
<td>698</td>
<td>1671</td>
<td>4985</td>
<td>-</td>
</tr>
<tr>
<td>GHM</td>
<td>8</td>
<td>47.0</td>
<td>4841</td>
<td>6138.7</td>
<td>3093</td>
<td>263</td>
<td>1517</td>
<td>1639</td>
<td>8039</td>
<td>n.r.</td>
</tr>
</tbody>
</table>

n.r. not reported

3.4. Conclusions

A new mathematical formulation has been developed for batch production scheduling in which sequence dependent changeover costs were considered. The resources required for changeover are considered implicitly in the form of changeover costs. Also catered for, are utility constraints, NIS policy and the storage of intermediates in idle units. The effectiveness of the proposed formulation has been demonstrated with a case study. In obtaining the optimum objective value, the formulation resulted in a smaller model (fewer binary variables, continuous variables and constraints) compared to an existing formulation.

In the following chapter, the scheduling formulation with sequence dependent changeover is used to address wastewater generation in batch plants.
Chapter 3. A New Scheduling Formulation for Multipurpose Batch Plants with Sequence Dependent Changeover

Nomenclature

Sets

$J = \{ j | j = \text{a unit} \}$

$J_s = \{ j_s | j_s = \text{a unit producing state } s \}$

$J_{st} = \{ j_{st} | j_{st} = \text{a unit that can store state } s \}$

$P = \{ p | p = \text{a time point} \}$

$S = \{ s | s = \text{any state other than a product} \}$

$S^p = \{ s^p | s^p = \text{a product state} \}$

$S^{mis} = \{ s^{mis} | s^{mis} = \text{states with no intermediate storage} \}$

$S^{st} = \{ s^{st} | s^{st} = \text{states which can be stored in a unit} \}$

$S_{in,j} = \{ s_{in,j} | s_{in,j} = \text{tasks performed in unit } j \}$

$S^{sc}_{in,j} = \{ s^{sc}_{in,j} | s^{sc}_{in,j} = \text{tasks which consume state } s \}$

$S^{sp}_{in,j} = \{ s^{sp}_{in,j} | s^{sp}_{in,j} = \text{tasks which produce state } s \text{ other than a product} \}$

$S^*_{in,j} = \{ s^*_{in,j} | s^*_{in,j} = \text{effective state representing a task} \}$

$S^{s,p}_{in,j} = \{ s^{s,p}_{in,j} | s^{s,p}_{in,j} = \text{tasks which produce state } s \text{ that is a product} \}$

$S^u_{in,j} = \{ s^u_{in,j} | s^u_{in,j} = \text{tasks which require utility for processing to occur} \}$

$u = \{ u | u = \text{a utility} \}$

Continuous variables

$L(u, p) = \text{level of utility } u \text{ at time point } p$

$m_s(s_{in,j}, p) = \text{amount of material processed by a task at time point } p$

$q_s(s, p) = \text{amount of state } s \text{ stored in a storage unit at time point } p$

$q_s(s, j, p) = \text{amount of state } s \text{ stored in unit } j \text{ at time point } p$
Chapter 3. A New Scheduling Formulation for Multipurpose Batch Plants with Sequence Dependent Changeover

\[ Q(s_{m,j}, u, p) \] amount of utility \( u \) required for processing \( s_{m,j} \)

\[ t_p(s_{m,j}, p) \] time at which a state is produced from unit \( j \) at time point \( p \)

\[ t_u(s_{m,j}, p) \] time at which a state is used in unit \( j \) at time point \( p \)

\[ tw_{out}(s_{m,j}, p) \] finish time of the changeover operation after task \( s_{m,j} \) has taken place

\[ u(s_{m,j}, p) \] amount of material stored in unit \( j \) that produced it at time point \( p \)

Binary variables

\[ t(j, p) \] binary variable associated with the usage of intermediate state produced by unit \( j \) at time point \( p \)

\[ x(s, p) \] binary variable associated with the availability of storage for state \( s \) at time point \( p \)

\[ x_{ch}(s_{m,j}, s'_{m,j}, p) \] binary variable associated with changeover from \( s_{m,j} \) to \( s'_{m,j} \) at time point \( p \)

\[ x_u(s, j, p) \] binary variable associated with storage of state \( s \) in unit \( j \) at time point \( p \)

\[ y(s_{m,j}, p) \] binary variable associated with usage of state \( s \) in unit \( j \) at time point \( p \)

\[ z(j, j', p) \] binary variable indicating that the amount of utility made available from \( j' \) is used immediately in unit \( j \)

Parameters

\[ \alpha(s_{m,j}) \] constant coefficient of processing time of a task

\[ \beta(s_{m,j}) \] variable coefficient of processing time of a task

\[ CCH(s_{m,j}, s'_{m,j}) \] cost associated with changeover from \( s_{m,j} \) to \( s'_{m,j} \)

\[ CP(s) \] cost price of product \( s \)

\[ CR(s) \] cost price of raw material \( s \)

\[ f(s_{m,j}) \] constant amount of utility required by a task

© University of Pretoria
Chapter 3. A New Scheduling Formulation for Multipurpose Batch Plants with Sequence Dependent Changeover

\[ g(s_{m,j}) \] variable amount of utility required by a task

\[ H \] time horizon of interest

\[ Q^U(u) \] upper bound on the amount of utility \( u \)

\[ V^U_j \] maximum capacity of unit \( j \)

\[ V^U_{s_{m,j}} \] maximum capacity of unit \( j \) to process a particular task

\[ V^L_{s_{m,j}} \] minimum capacity of unit \( j \) to process a particular task

\[ \delta_{s_{m,j}}^{sc} \] portion of state \( s \) consumed by a task from intermediate storage

\[ \delta_{s_{m,j}}^{sp} \] portion of state \( s \) produced by a task stored in intermediate storage

\[ \gamma_{s_{m,j}}^{sc} \] portion of state \( s \) consumed by a task from idle processing unit

\[ \gamma_{s_{m,j}}^{sp} \] portion of state \( s \) produced by a task stored in idle processing unit

\[ \rho_{s_{m,j}}^{sc} \] portion of state \( s \) consumed by a task

\[ \rho_{s_{m,j}}^{sp} \] portion of state \( s \) produced by a task

\[ \tau_{ch}(s_{m,j}, s'_{m,j}) \] duration of changeover when \( s_{m,j} \) is processed before \( s'_{m,j} \) (sequence dependent)

References


CHAPTER 4
Wastewater Minimisation in Batch Plants with Sequence Dependent Changeover

Introduction

Wastewater from washing contributes a considerable part to the total amount of wastewater in batch plants, due to sharing of equipment and the requirement to ensure product integrity. The objective of the work presented in this chapter is to develop a mathematical model for the simultaneous optimisation of batch production scheduling and water use in a multipurpose batch plant in which the water requirement is determined by the sequence of tasks in units.

4.1. Motivation for the study

From the previous chapter, sequence dependent changeover costs were determined due to the ability of the presented scheduling formulation to adequately identify the unique successors of tasks occurring in units. Regarding wastewater minimisation, it has already been established from the literature review that a true minimum for water usage in batch plants can only be obtained if production scheduling and water use optimisation are performed simultaneously. In the published literature, water requirement is sequence independent. Missing from literature is a formulation that determines the water requirement in a unit based on a task and its unique successor in the unit (sequence dependent).

The illustration in Figure 4.1 explains the significance of the contribution proposed in this work. The figure depicts the resulting production schedules of a plant with 3 tasks that can take place in any of 2 units, over a 5.5 h time horizon. The fixed durations of Task A, Task B and Task C are 2 h, 0.8 h and 2 h respectively. The fixed washing durations after Task A, Task B and Task C have taken place in either unit are 0.5 h, 0.2 h and 1.0 h respectively. Specifically, Figure 4.1(a) represents the production schedule obtainable for the case of sequence independent washing, where the full extent of washing is required after a task is completed, irrespective of the task that will succeed it. Figure 4.1(b) represents the
production schedule obtainable using the formulation proposed in this contribution, that is, sequence dependent washing. Here, washing is performed after a task occurs depending on the task that will succeed it. In the proposed formulation, a condition stating that a unit need not be washed if a task succeeds another can be incorporated. Similarly, a condition stating that only a degree of washing instead of the full extent is necessary when a task succeeds another can be catered for.

In the motivating example, a unit need not be washed if Task C succeeds itself. Secondly, only a degree of washing is required when Task B succeeds Task C on a unit. Observed from Figure 4.1 (b), the first and second conditions have been exploited. Task C succeeds itself in unit 2; therefore, the unit need not be washed. Task B succeeds Task C in unit 2, therefore, only a degree of washing is necessary. Furthermore, Task B which hitherto was not performed for a lack of available equipment to process it within the time horizon of 5.5 h in Figure 4.1(a) is now processed in Figure 4.1(b). Consequently, the benefits are twofold, a reduction in freshwater use and wastewater generation is observed together with an increase in production capacity with a potential to contribute to the overall profit.

From the above discussion, it is clear that the order in which tasks take place in a unit affects freshwater use, wastewater generation and overall profit.
4.2. Problem statement

The problem considered in this chapter can be stated formally as follows.

Given the following scheduling data:
(i) production recipe including processing times of operations required to convert raw materials to product,
(ii) available units and their capacities,
(iii) availability and capacity of storage for each material,
(iv) time horizon of interest,
(v) costs of raw materials and products
(vi) selling price of final products.

and wastewater minimisation data:
(i) mass load of contaminant,
(ii) maximum inlet and outlet concentration of contaminants,
(iii) sequence dependent changeover time,
(iv) cost of freshwater and effluent treatment.

Determine the production schedule that achieves the maximum profit through reduced freshwater use while taking into consideration the sequence of tasks.

4.3. Mathematical formulation

The sets, parameters and variables on which the mathematical formulation is based can be found in the nomenclature before the end of this chapter.

The mathematical formulation consists of two modules, i.e. batch production scheduling and wastewater minimisation modules. The constraints of the formulation are based on the superstructure in Figure 4.2. The figure depicts three different changeover scenarios that are possible in unit \( j \) after the completion of task \( s_{m,j} \), to prepare the unit for a succeeding task from the set of tasks, \( S_{m,j} \). The full extent of washing is required in the first scenario to prepare the unit for task \( s'_{m,j} \). In the second scenario, only a degree of washing is necessary to
Chapter 4. Wastewater Minimisation in Batch Plants with Sequence Dependent Changeover

prepare the unit for the succeeding task, $s''_{m,j}$. In the third scenario, no washing is required to prepare the unit for the succeeding task, $s''_{m,j}$.

![Diagram](image)

Figure 4.2. Superstructure for mathematical formulation

4.3.1. Scheduling constraints

Constraints of the scheduling formulation incorporating sequence dependent changeover were presented in Chapter 3. An important feature still needs clarification, the identification of the last task in a unit. From the production schedule in Figure 3.2, no changeover task was required after the completion of the last task in each unit. This is by virtue of the changeover variable, $x_{ch}(s_{m,j}, s'_{m,j}, p)$ taking a value of 0, since no task succeeds the last task. It is good practice to ensure that all units are cleaned at the end of a production shift.

4.3.1.1. Last task in a unit

The last task in a unit does not necessarily occur at the last time point, i.e., $p = |P|$. To ensure that no unit is left unwashed at the end of the time horizon, a binary variable, $XL(s_{m,j}, p)$ is defined. This binary variable is equal to 1 if task $s_{m,j}$ taking place at time point $p$ is the last task in the unit. Constraint (4.1) captures the condition that at most one task will be the last to be performed in a unit, within the time horizon of interest.
\[ \sum_{p} \sum_{s_{in,j} \in S_{in,j}} XL(s_{in,j}, p) \leq 1 \quad \forall \; j \in J \] (4.1)

Constraint (4.2) states that if a task is the last in a unit, no other task can be performed after it. The constraint works as follows: if a task takes place in a unit, \( y(s_{in,j}, p) = 1 \). If no other task succeeds this, \( \sum_{s_{in,j} \in S_{in,j}} x_{ch}(s_{in,j}, s_{in,j}, p) = 0 \). This forces the value of \( XL(s_{in,j}, p) \) to be 1.

\[ \sum_{s_{in,j} \in S_{in,j}} x_{ch}(s_{in,j}, s_{in,j}, p) + XL(s_{in,j}, p) = y(s_{in,j}, p) \]
\[ \forall \; j \in J, \; s_{in,j} \in S_{in,j}, \; p \in P \] (4.2)

**4.3.2. Wastewater minimisation constraints**

After a processing task has occurred in a unit, the unit may be washed to prepare it for another processing task.

**4.3.2.1. Mass balance constraints**

The water inlet to a unit is freshwater, depicted by Constraint (4.3). Constraint (4.4) states that water leaving a unit is discarded as effluent. Constraint (4.5) states that water is neither produced nor lost during washing. The amount of water entering a unit is equal to the amount of water leaving the unit.

\[ F_{in}(s_{in,j}, p) = F_{j}(s_{in,j}, p) \quad \forall \; j \in J, \; s_{in,j} \in S_{in,j}, \; p \in P \] (4.3)

\[ F_{out}(s_{in,j}, p) = F_{j}(s_{in,j}, p) \quad \forall \; j \in J, \; s_{in,j} \in S_{in,j}, \; p \in P \] (4.4)

\[ F_{in}(s_{in,j}, p) = F_{out}(s_{in,j}, p) \quad \forall \; j \in J, \; s_{in,j} \in S_{in,j}, \; p \in P \] (4.5)

Constraint (4.6) determines the mass load in a unit after a state has been processed. It is a product of the contaminant loading and the amount of material processed. Constraint (4.7) calculates the outlet contaminant mass from a unit as the mass load of contaminant picked up in the unit during washing. In Constraint (4.7), the first term on the right-hand side of the
equation caters for the situation whereby the processing task just completed, has a processing task succeeding it. The second term caters for the situation whereby the processing task just completed is the last one in the unit. The inlet contaminant mass is zero since freshwater is used.

\[
M_j = \Delta M_s \cdot m_u s_i \quad \forall \quad j \in J, \quad s_{i,j} \in S_{i,j}, \quad c \in C, \quad p \in P 
\]  

(4.6)

\[
F_{out_j} p_{out_i} = \sum_{s_{i,j} \in S_{i,j}} \left( M_j \cdot x_{ch} s_i, s_i' \cdot p \right) + M_j \cdot XL s_i \quad \forall \quad j \in J, \quad s_{i,j} \in S_{i,j}, \quad c \in C, \quad p \in P 
\]  

(4.7)

Constraint (4.8) ensures that the outlet contaminant concentration does not exceed the allowed maximum, whether the associated processing task has a succeeding task or is the last task in the unit. Constraint (4.9) ensures that the maximum allowable water mass in a unit is not exceeded. This maximum is calculated in Equation (4.10) as explained by Majozi and Gouws (2009).

\[
c_{out_i} s_{i,j} \leq C_{out_i} s_{i,j} \left( \sum_{s_{i,j} \in S_{i,j}} x_{ch} s_i, s_i' \cdot p \right) + XL s_i \quad \forall \quad j \in J, \quad s_{i,j} \in S_{i,j}, \quad c \in C, \quad p \in P 
\]  

(4.8)

\[
F_{in_i} s_{i,j} \leq F_{in_i} s_{i,j} \left( \sum_{s_{i,j} \in S_{i,j}} x_{ch} s_i, s_i' \cdot p \right) + XL s_i \quad \forall \quad j \in J, \quad s_{i,j} \in S_{i,j}, \quad p \in P 
\]  

(4.9)

\[
F_{in_i} s_{i,j} = \max_{c \in C} \left\{ \frac{M_j s_{i,j} c}{C_{out_i} s_{i,j} c - C_{in_i} s_{i,j} c} \right\} \quad \forall \quad j \in J, \quad s_{i,j} \in S_{i,j} 
\]  

(4.10)
4.3.2.2. Task scheduling constraints

The task scheduling constraints ensure the proper sequencing of equipment washing and production tasks. According to Constraint (4.2) above, at a given time point, a task either has a successor or is the final task in the unit. As a result, the washing duration is determined by which binary variable, either \( x_{ch}(s_{in,j}, s'_{in,j}, p) \) or \( XL(s_{in,j}, p) \) is 1. Constraint (4.11) determines the finish time of the washing operation and implies that unit \( j \) is washed immediately after a production task has ended in the unit depending on the succeeding production task that may take place in the unit. Constraint (4.12) states that the start time of the succeeding production task in unit \( j \) is after the completion of washing in the same unit. Constraint (4.13) ensures that the finish time of washing takes place within the time horizon.

\[
w_{out}(s_{in,j}, p) = t_p(s_{in,j}, p) + \sum_{s_{in,j}} \left( \tau_{ch}(s_{in,j}, s'_{in,j}) \cdot x_{ch}(s_{in,j}, s'_{in,j}, p) \right) + \tau_w(s_{in,j}) \cdot XL(s_{in,j}, p) \\
\forall \ j \in J, \ s_{in,j} \in S_{in,j}, \ p \in P \tag{4.11}
\]

\[
t_s(s'_{in,j}, p) \geq w_{out}(s_{in,j}, p') - H(1 - x_{ch}(s_{in,j}, s'_{in,j}, p')) \\
\forall \ j \in J, \ s_{in,j}, s'_{in,j} \in S_{in,j}, \ p, p' \in P, \ p' < p, p' < P \tag{4.12}
\]

\[
w_{out}(s_{in,j}, p) \leq H \quad \forall \ j \in J, \ s_{in,j} \in S_{in,j}, \ p \in P
\]

(4.13)

4.3.3. Water reuse

The constraints discussed so far considered the use of freshwater for washing units depending on the sequence of tasks. The effect of direct water reuse as an additional option for water use is now investigated and is based on the superstructure in Figure 4.3.
4.3.3.1. Mass balance constraints

Constraint (4.14) states that the water inlet to a unit could consist of freshwater and reuse water from other units. Constraint (4.15) states that the water outlet could be discharged as effluent or reused directly by other available units. Constraint (4.5) remains the same as no water is lost during washing. Constraint (4.6) calculates the contaminant mass load as before. Constraint (4.16) determines the outlet contaminant mass from a unit after washing. The constraint includes the mass of contaminant that is present in reuse water. Constraint (4.17) limits the amount of water reused to the maximum allowable water mass of the receiving unit.

\[
F_{in}(s_{in,j}, p) = F_j(s_{in,j}, p) + \sum_{s_{n,j} \in S_{n,j}} F_i(s_{in,j}, s_{in,j}, p-1)
\]
\[
\forall \ j, j' \in J, \ s_{in,j} \in S_{in,j}, \ p \in P, \ p > p_i \quad (4.14)
\]

\[
F_{out}(s_{in,j}, p) = F_j(s_{in,j}, p) + \sum_{s_{n,j} \in S_{n,j}} F_i(s_{in,j}, s_{in,j}, p)
\]
\[
\forall \ j, j' \in J, \ s_{in,j} \in S_{in,j}, \ p \in P \quad (4.15)
\]

\[
F_{out}(s_{in,j}, p)c_{out}(s_{in,j}, c, p) = \sum_{s_{m,j} \in S_{m,j}} \left(M(s_{m,j}, c, p) \cdot x_{ch}(s_{m,j}, s'_{m,j}, p)\right) + M(s_{in,j}, c, p) \cdot XL(s_{in,j}, p)
\]
\[
+ \sum_{s_{n,j} \in S_{n,j}} F_i(s_{in,j}, s_{in,j}, p-1) \cdot c_{out}(s_{in,j}, p)
\]
\[
\forall \ j, j' \in J, \ s_{in,j} \in S_{in,j}, \ c \in C, \ p \in P, \ p > p_i \quad (4.16)
\]
Chapter 4. Wastewater Minimisation in Batch Plants with Sequence Dependent Changeover

$$F_r(s_{m,j}, s_{m,j}, p) \leq F^{E_i}(s_{m,j}) \cdot y_r(s_{m,j}, s_{m,j}, p)$$
$$\forall \ j, j' \in J, \ s_{m,j}, s_{m,j} \in S_{m,j}, \ p \in P$$

(4.17)

4.3.3.2. Sequencing constraints for direct water reuse

Previously, it was assumed that the start time of washing coincided with the finish time of the production task. To improve the opportunities for direct reuse, it is now assumed that the start time of washing can take place at any time after the completion of the production task. This is expressed in Constraint (4.18). Constraint (4.19) determines the finish time of washing. Constraint (4.12) remains valid and ensures that the start time of the succeeding production task in the unit is after the completion of washing as a result of the previous production task. Constraints (4.20) and (4.21) are similar to Constraint (4.13) and ensure that the start time of washing and the time at which direct water reuse takes place, occur within the time horizon respectively.

$$t_{w_{in}}(s_{m,j}, p) \geq t_{p}(s_{m,j}, p) \ \ \forall \ j \in J, \ s_{m,j} \in S_{m,j}, \ p \in P$$

(4.18)

$$t_{w_{out}}(s_{m,j}, p) = t_{w_{in}}(s_{m,j}, p) + \sum_{s_{m,j}}(\tau_{ch}(s_{m,j}, s_{m,j}^{'}) \cdot x_{ch}(s_{m,j}, s_{m,j}^{'}, p)) + \tau_{w}(s_{m,j})\cdot X_{L}(s_{m,j}, p)$$
$$\forall \ j \in J, \ s_{m,j} \in S_{m,j}, \ p \in P$$

(4.19)

$$t_{w_{in}}(s_{m,j}, p) \leq H \ \ \forall \ j \in J, \ s_{m,j} \in S_{m,j}, \ p \in P$$

(4.20)

$$t_{w}(s_{m,j}, s_{m,j}, p) \leq H \ \ \forall \ j, j' \in J, \ s_{m,j}, s_{m,j} \in S_{m,j}, \ p \in P$$

(4.21)

Constraints (4.22) and (4.23) determine the relationship between the binary variable associated with direct reuse, $y_r(s_{m,j}, s_{m,j}, p)$, and the occurrence of washing which is dictated by the binary variables $x_{ch}(s_{m,j}, s_{m,j}^{'}, p)$ and $X_{L}(s_{m,j}, p)$. The constraints ensure $y_r(s_{m,j}, s_{m,j}, p)$ is 1 provided that the unit is washed either due to the succeeding production task or that the just completed task is the last one in the unit. It is also possible that no direct
reuse occurs even when the unit is washed. Constraint (4.22) is for the unit that sends reuse water on completion of washing while (4.23) is for the unit receiving reuse water at the beginning of washing.

\[
y_r(s_{m,j}, s_{m,j}', p) \leq XL(s_{m,j}, p) + \sum_{s'_{m,j} \in S_{m,j}} x_{ch}(s_{m,j}, s'_{m,j}, p)
\]
\[\forall \ j, j' \in J, \ s_{m,j}, s_{m,j}' \in S_{m,j}, \ p \in P\] (4.22)

\[
y_r(s_{m,j}, s_{m,j}', p-1) \leq XL(s_{m,j}, p) + \sum_{s'_{m,j} \in S_{m,j}} x_{ch}(s_{m,j}', s_{m,j}', p)
\]
\[\forall \ j, j' \in J, \ s_{m,j}, s_{m,j}' \in S_{m,j}, \ p \in P, p > p_1\] (4.23)

Constraints (4.24) and (4.25) together ensure that the time at which direct reuse occurs coincides with the finish time of washing for the unit that is discharging water. Similarly, Constraints (4.26) and (4.27) together ensure that the time at which direct reuse occurs coincides with the start time of washing for the unit that is receiving water.

\[
tw_r(s_{m,j}, s_{m,j}', p) \geq tw_{out}(s_{m,j}, p) - H\left(1 - y_r(s_{m,j}, s_{m,j}', p)\right)
\]
\[\forall \ j, j' \in J, \ s_{m,j}, s_{m,j}' \in S_{m,j}, \ p \in P\] (4.24)

\[
tw_r(s_{m,j}, s_{m,j}', p) \leq tw_{out}(s_{m,j}, p) + H\left(1 - y_r(s_{m,j}, s_{m,j}', p)\right)
\]
\[\forall \ j, j' \in J, \ s_{m,j}, s_{m,j}' \in S_{m,j}, \ p \in P\] (4.25)

\[
tw_r(s_{m,j}, s_{m,j}', p-1) \geq tw_{in}(s_{m,j}', p) - H\left(1 - y_r(s_{m,j}, s_{m,j}', p-1)\right)
\]
\[\forall \ j, j' \in J, \ s_{m,j}, s_{m,j}' \in S_{m,j}, \ p \in P, p > p_1\] (4.26)

\[
tw_r(s_{m,j}, s_{m,j}', p-1) \leq tw_{in}(s_{m,j}', p) + H\left(1 - y_r(s_{m,j}, s_{m,j}', p-1)\right)
\]
\[\forall \ j, j' \in J, \ s_{m,j}, s_{m,j}' \in S_{m,j}, \ p \in P, p > p_1\] (4.27)
4.3.4. Objective function

The objective is to maximise profit while minimising the amount of freshwater required. Equation (4.28) is the revenue from product, Equation (4.29) is the cost of raw material and Equation (4.30) is the cost of freshwater and effluent treatment. The objective in Equation (4.31) is maximized.

\[
\text{Revenue} = \sum_{s \in P} q_s(s, p)CP(s) \quad \forall \ p = P \tag{4.28}
\]

\[
\text{Costraw} = \sum_{s_{u,n} \in S_{u,n}} \sum_{p \in P} CR(s)\rho_{u} m_u(s_{m,j}, p) \tag{4.29}
\]

\[
\text{Costwater} = CFW \sum_{s_{u,n} \in S_{u,n}} \sum_{p \in P} F_j(s_{m,j}, p) + CFE \sum_{s_{u,n} \in S_{u,n}} \sum_{p \in P} F_e(s_{m,j}, p) \tag{4.30}
\]

\[
\text{Objective} = \text{Revenue} - \text{Costraw} - \text{Costwater} \tag{4.31}
\]

Constraints (4.7) and (4.16), the contaminant mass balance for the freshwater model and direct water reuse model, respectively feature products of continuous and binary variables. These cause nonlinearity in the model and can be linearised using Glover (1975) transformation. Furthermore, if a problem involves a single contaminant, the outlet contaminant concentration could be set to the maximum according to Savelski and Bagajewicz (2000). In such a case, with the appropriate substitutions, the resulting formulation is a mixed integer linear problem, MILP.

4.4. Case studies

Three case studies are examined; the first and third consider the same multiproduct facility and the second, a multipurpose facility. A single contaminant is assumed present in all cases. In order to demonstrate the influence of the task sequence on the amount of water required for washing, three scenarios of case studies I and II are observed.
Scenario 1: Freshwater is used for washing without considering the sequence of tasks. After a task is performed, the unit is washed so that it is suitable for any future tasks. As such, the washing duration of a unit after a task has taken place is the maximum of the changeover times required for every possible succeeding task. Furthermore, the entire mass load of contaminant after a task has been performed must be picked up by water.

Scenario 2: Freshwater is used for washing while considering the sequence of tasks. After a task is performed, the unit is washed so that it is suitable particularly for the succeeding task. The washing duration of a unit after a task has taken place is determined by the specific succeeding task. The entire mass load of contaminant is picked up by water.

Scenario 3: Freshwater is used for washing while considering the sequence of tasks. After a task is performed, the unit is washed so that it is suitable particularly for the succeeding task. The washing duration of a unit after a task has taken place is determined by the specific succeeding task. A fraction of the mass load of contaminant is picked up by water, obtained by applying Constraint (4.32) instead of Constraint (4.6). The assumption is that the mass load of contaminant picked up is proportional to the washing duration. The fraction of mass load of contaminant is approximated using the quotient of sequence dependent changeover time and maximum changeover time.

\[ M(s_{m,j}, c, p) = \Delta M(s_{m,j}, c) \cdot m_u(s_{m,j}, p) \left( \frac{\tau_{ch}(s_{m,j}, s_{m,j}^{'})}{\tau_{w}(s_{m,j})} \right) \]

\[ \forall \ j \in J, \ s_{m,j}, s_{m,j}^{'}, s_{m,j} \in S_{m,j}, \ c \in C, \ p \in P \]  

(4.32)

In order to demonstrate the influence of direct water reuse with appropriate task sequencing, four scenarios of case study III are investigated. Scenarios 1 and 2 are as described above.

Scenario 4: Freshwater and direct water reuse are available for washing while considering the sequence of tasks. This is Scenario 2 with direct water reuse constraints included.

Scenario 5: Freshwater and direct water reuse are available for washing without considering the sequence of tasks. This is Scenario 1 with direct water reuse constraints included.
The models for case studies I and II consisted of Constraints (3.1)-(3.24), (4.1)-(4.13), (4.28)-(4.32) and were MILP models. The models for case study III consisted of Constraints (3.1)-(3.24), (4.1)-(4.2), (4.14)-(4.31) and were MILP models. The models were solved using GAMS 22.0 software on a machine with Intel(R) Core(TM) i7-2670QM, 2.2 GHz processor and 4.0GB RAM and CPLEX as the MILP solver.

4.4.1. Case study I

This case study from Maravelias and Grossmann (2003) considered sequence dependent changeover times (Shaik and Vooradi, 2013) and has been adapted to include water using aspects. Two products, P1 and P2 are to be fabricated from two raw materials, F1 and F2 in a production facility illustrated in Figure 4.4. The facility consists of two equipment units, U1 and U2 and intermediate storage tanks for each state. Table 4.1 provides the production processing data. The sequence dependent changeover times are listed in Table 4.2. Table 4.3 provides the water using requirements of the units after performing tasks. The objective is to maximise profit over a 16 h time horizon while fulfilling a minimum demand of 2 tons of each product. The cost of water and effluent treatment is $ 0.5/ kg and $ 0.25/ kg respectively. In Scenario 1, the washing durations of the units after tasks T11, T21, T12, T22, T13 and T23 are 0.5 h, 0.6 h, 0.3 h, 0.2 h, 0.6 h and 0.5 h, respectively and are the maximum changeover times in Table 4.2.

Figure 4.4. STN (a) and SSN (b) representation of case study I
Table 4.1. Production data for case study I

<table>
<thead>
<tr>
<th>Task</th>
<th>Unit</th>
<th>Min batch size (T)</th>
<th>Max batch size (T)</th>
<th>( \alpha ) (h)</th>
<th>( \beta ) (h/T)</th>
<th>Material state</th>
<th>Initial inventory (T)</th>
<th>Max storage (T)</th>
<th>Revenue/cost ($/T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T11</td>
<td>U1</td>
<td>2.0</td>
<td>5.0</td>
<td>0.5</td>
<td>0.40</td>
<td>s1</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>T21</td>
<td>U1</td>
<td>2.0</td>
<td>5.0</td>
<td>0.75</td>
<td>0.60</td>
<td>s2</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>T12</td>
<td>U2</td>
<td>1.2</td>
<td>3.0</td>
<td>1.0</td>
<td>1.33</td>
<td>s3</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>T22</td>
<td>U2</td>
<td>1.2</td>
<td>3.0</td>
<td>1.0</td>
<td>1.33</td>
<td>s4</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>T13</td>
<td>U1</td>
<td>2.0</td>
<td>5.0</td>
<td>0.5</td>
<td>0.40</td>
<td>s5</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>T23</td>
<td>U1</td>
<td>2.0</td>
<td>5.0</td>
<td>0.5</td>
<td>0.40</td>
<td>s6</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>s7</td>
<td>0</td>
<td>100</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>s8</td>
<td>0</td>
<td>100</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 4.2. Changeover times for case study I

<table>
<thead>
<tr>
<th></th>
<th>T11</th>
<th>T21</th>
<th>T12</th>
<th>T22</th>
<th>T13</th>
<th>T23</th>
</tr>
</thead>
<tbody>
<tr>
<td>T11</td>
<td>0.0</td>
<td>0.2</td>
<td>0.1</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T21</td>
<td>0.3</td>
<td>0.0</td>
<td>0.6</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T12</td>
<td>0.0</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T22</td>
<td>0.2</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T13</td>
<td>0.2</td>
<td>0.6</td>
<td>0.0</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T23</td>
<td>0.5</td>
<td>0.3</td>
<td>0.4</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3. Water using data for case study I

<table>
<thead>
<tr>
<th>Task</th>
<th>Unit</th>
<th>Max inlet concentration (ppm)</th>
<th>Max outlet concentration (ppm)</th>
<th>Contaminant loading (g contaminant/T batch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T11</td>
<td>U1</td>
<td>500</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>T21</td>
<td>U1</td>
<td>500</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>T12</td>
<td>U2</td>
<td>1000</td>
<td>2000</td>
<td>100</td>
</tr>
<tr>
<td>T22</td>
<td>U2</td>
<td>1000</td>
<td>2000</td>
<td>100</td>
</tr>
<tr>
<td>T13</td>
<td>U1</td>
<td>500</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>T23</td>
<td>U1</td>
<td>500</td>
<td>1000</td>
<td>100</td>
</tr>
</tbody>
</table>

4.4.1.1. Results and discussion

The results of Scenarios 1-3 are illustrated with their respective production schedules in Figures 4.5-4.7 respectively. The clear blocks represent the unit undergoing production while the shaded blocks represent washing. The numbers in brackets represent the amount of material being processed and the numbers below the washing operations represent freshwater. In Scenario 1, the revenue achieved was $7007 and the amount of freshwater used was 1751.8 kg, resulting in a profit of $5693.2. From the production schedule in Figure 4.5, it can
be observed that unit U2 is a critical unit. The processing durations are longer in U2 compared to U1 even when the amounts of material processed in both units are similar.

**Figure 4.5. Scenario 1, case study I**

The introduction of sequence dependent changeover in *Scenario 2* returned revenue of $\$7155.2$ with a profit of $\$6151$ and associated water usage of $1338.8$ kg. Compared to *Scenario 1*, the objective improved by $8\%$ and the water usage reduced by $23.5\%$. From Figure 4.6, this improvement was due to the elimination of the washing task before performing T22 in U2 at time point $p4$ since no washing is required when the same task is performed successively in a unit (Table 4.2). This translates to an increase in the time available for U2 to process materials within the time horizon of interest. In this case study, processing duration is a function of the amount of material processed; therefore, U2 can process more material in this scenario. U2 processes a total amount of $7.16$ kg in *Scenario 2* compared to $7.01$ kg in *Scenario 1*. This is responsible for the increased revenue obtained in *Scenario 2*. The savings in water is evident, as there is no washing between T22 at $p3$ and T22 at $p4$ in U2. Also, washing is not required between T23 at $p4$ and T23 at $p5$ in U1. These together led to an improvement in the objective.
Figure 4.6. Scenario 2, case study I

The revenue achieved in Scenario 3 was $7155.2, the profit was $6275 and the water usage was 1173.5 kg. The production schedules of Scenario 2 in Figure 4.6 and Scenario 3 in Figure 4.7 are similar except for a difference in the amount of freshwater used. The mass load of contaminant picked up during washing in Scenario 3 is dependent on the succeeding task. The total mass load of contaminant picked up in Scenario 3 is less than in Scenario 2 (see Equation 4.32). Hence, the amount of washing water required is reduced. Compared to Scenario 1, the objective improved by 10% and water usage reduced by 33%. The computational statistics of the scenarios are presented in Table 4.4.

Figure 4.7. Scenario 3, case study I

Table 4.4. Computational statistics of case study I

<table>
<thead>
<tr>
<th>Scenario</th>
<th>P</th>
<th>CPU time (s)</th>
<th>Nodes</th>
<th>RMILP</th>
<th>MILP</th>
<th>B.V.</th>
<th>C.V.</th>
<th>Constraints</th>
<th>Non-zeros</th>
<th>Relative gap %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>0.10</td>
<td>75</td>
<td>9000</td>
<td>5693</td>
<td>66</td>
<td>520</td>
<td>1050</td>
<td>2482</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0.61</td>
<td>92</td>
<td>9000</td>
<td>6151</td>
<td>176</td>
<td>816</td>
<td>1918</td>
<td>6608</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.52</td>
<td>114</td>
<td>9000</td>
<td>6275</td>
<td>176</td>
<td>816</td>
<td>1918</td>
<td>6608</td>
<td>-</td>
</tr>
</tbody>
</table>
4.4.2. Case study II

This case study from Kondili et al. (1993) has been adapted to consider sequence dependent changeover and water using operations. Two products, Product 1 and Product 2 are to be produced from raw materials, Feed A, Feed B and Feed C. Figure 4.8 provides the STN and SSN representations of the problem. Four units are available, a heater, HR, two reactors, RR1 and RR2 and a separator, SR. HR is suitable for heating Feed A, while SR is suitable for purifying Impure E. RR1 and RR2 are suitable for performing Reaction 1, R1, Reaction 2, R2 and Reaction 3, R3. The objective is to maximise profit while minimising the amount of water used for washing. The time horizon of interest is 10 h. The production data are detailed in Table 4.5. The sequence dependent changeover times are provided in Table 4.6 and the water requirements for this case study are provided in Table 4.7. In Scenario 1, the washing durations of the units after tasks R1-RR1, R1-RR2, R2-RR1, R2-RR2, R3-RR1 and R3-RR2 are 0.25 h, 0.3 h, 0.25 h, 0.3 h, 0.25 h and 0.3 h, respectively and are the maximum changeover times in Table 4.6.

Figure 4.8. STN (a) and SSN (b) representation of case study II

© University of Pretoria
Table 4.5. Data pertaining to production for case study II (Halim and Srinivasan 2012)

<table>
<thead>
<tr>
<th>Task</th>
<th>Unit</th>
<th>Max batch size (kg)</th>
<th>α (h)</th>
<th>β (h)</th>
<th>Material state</th>
<th>Initial inventory (kg)</th>
<th>Max storage (kg)</th>
<th>Revenue/cost ($/kg or $/MJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heating (H)</td>
<td>HR</td>
<td>100</td>
<td>0.667</td>
<td>0.007</td>
<td>Feed A (s1)</td>
<td>1000</td>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td>Reaction 1 (R1)</td>
<td>RR1</td>
<td>50</td>
<td>1.334</td>
<td>0.027</td>
<td>Feed B (s2)</td>
<td>1000</td>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>80</td>
<td>1.334</td>
<td>0.017</td>
<td>Feed C (s3/s4)</td>
<td>1000</td>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td>Reaction 2 (R2)</td>
<td>RR1</td>
<td>50</td>
<td>1.334</td>
<td>0.027</td>
<td>Hot A (s5)</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>80</td>
<td>1.334</td>
<td>0.017</td>
<td>Int AB (s8)</td>
<td>0</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>Reaction 3 (R3)</td>
<td>RR1</td>
<td>50</td>
<td>0.667</td>
<td>0.013</td>
<td>Int BC (s6)</td>
<td>0</td>
<td>150</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>80</td>
<td>0.667</td>
<td>0.008</td>
<td>Impure E (s9)</td>
<td>0</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>Separation (S)</td>
<td>SR</td>
<td>200</td>
<td>1.334</td>
<td>0.007</td>
<td>Product 1 (s7)</td>
<td>0</td>
<td>1000</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Product 2 (s10)</td>
<td>0</td>
<td>1000</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Washwater</td>
<td>0</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Wastewater</td>
<td>0</td>
<td></td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 4.6. Changeover times for case study II

<table>
<thead>
<tr>
<th>Changeover</th>
<th>R1-RR1</th>
<th>R1-RR2</th>
<th>R2-RR1</th>
<th>R2-RR2</th>
<th>R3-RR1</th>
<th>R3-RR2</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1-RR1</td>
<td>0.25</td>
<td>0.2</td>
<td></td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1-RR2</td>
<td>0.25</td>
<td>0.25</td>
<td>0.15</td>
<td></td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>R2-RR1</td>
<td>0.25</td>
<td>0.3</td>
<td></td>
<td>0.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2-RR2</td>
<td>0.25</td>
<td>0.3</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R3-RR1</td>
<td>0.25</td>
<td>0.3</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.7. Data pertaining to water requirements for case study II

<table>
<thead>
<tr>
<th>Task</th>
<th>Unit</th>
<th>Max inlet concentration (ppm)</th>
<th>Max outlet concentration (ppm)</th>
<th>Contaminant loading (g contaminant/kg batch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reaction 1 (R1)</td>
<td>RR1</td>
<td>300</td>
<td>700</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>300</td>
<td>700</td>
<td>0.2</td>
</tr>
<tr>
<td>Reaction 1 (R1)</td>
<td>RR1</td>
<td>700</td>
<td>1000</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>700</td>
<td>1000</td>
<td>0.2</td>
</tr>
<tr>
<td>Reaction 1 (R1)</td>
<td>RR1</td>
<td>500</td>
<td>800</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>500</td>
<td>800</td>
<td>0.2</td>
</tr>
</tbody>
</table>

4.4.2.1. Results and discussion

The resulting production schedule of Scenario 1 is illustrated in Figure 4.9. The objective achieved was $ 3472.6 from the obtained revenue of $ 3488.3 and freshwater usage of 104.8 kg.
The resulting production schedule of Scenario 2 is provided in Figure 4.10. In this scenario, the objective was $3566.7 with associated revenue of $3577.5 and freshwater usage of 72.3 kg. This represents a 2.70% increase in profit and reduction in freshwater requirement of 31%. As shown in Figure 4.10, the introduction of sequence dependent changeover gives rise to a production schedule that favors the consecutive processing of the same task in a unit. Reaction R1 in RR1 occurs at time point $p_2$ and again at $p_3$ and $p_4$, in this way, changeover is not necessary (see Table 4.6). Similarly, reaction R3 in RR2 occurs at time point $p_3$ and follows immediately at $p_4$. Furthermore, the increased revenue in Scenario 2 was due to a slight increase in the total amount of product compared to Scenario 1. The amounts of Product 1 and Product 2 in Scenario 2 were 84.14 kg and 94.74 kg respectively, while, the amounts of Product 1 and Product 2 in Scenario 1 were 86.67 kg and 87.75 kg respectively. These factors are responsible for the increase in the objective value, i.e. profit.
The resulting production schedule of Scenario 3 is provided in Figure 4.11. In Scenario 3, the objective was $3569.4 with achieved revenue of $3577.5 and freshwater use of 53.7 kg. This is a 2.78% improvement in profit over Scenario 1 with a water reduction of 48.8%. The revenue achieved in Scenarios 2 and 3 are exactly the same but the amount of water required for washing is reduced in Scenario 3 (due to the reduced contaminant mass load). This is solely responsible for the improvement of the objective in Scenario 3 over Scenario 2. The computational statistics of the scenarios are presented in Table 4.8.
Table 4.8. Computational statistics of case study II

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$P$</th>
<th>CPU time (s)</th>
<th>Nodes</th>
<th>RMILP</th>
<th>MILP</th>
<th>B.V.</th>
<th>C.V.</th>
<th>Constraints</th>
<th>Non-zeros</th>
<th>Relative gap %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>0.13</td>
<td>0</td>
<td>3841.1</td>
<td>3472.6</td>
<td>56</td>
<td>495</td>
<td>991</td>
<td>2516</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>1.29</td>
<td>965</td>
<td>4818.5</td>
<td>3566.7</td>
<td>174</td>
<td>800</td>
<td>1975</td>
<td>6615</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>1.01</td>
<td>663</td>
<td>4821.0</td>
<td>3569.5</td>
<td>174</td>
<td>800</td>
<td>1975</td>
<td>6615</td>
<td>-</td>
</tr>
</tbody>
</table>

4.4.3. Case study III

The batch facility investigated is the same as case study I, except with different water using data provided in Table 4.9. The main purpose of this case study is to highlight the difference between the production schedule obtained with direct water reuse and appropriate task sequencing (for water use) and that obtained with direct water reuse when washing is sequence independent.

Table 4.9. Water using data for case study III.

<table>
<thead>
<tr>
<th>Task</th>
<th>Unit</th>
<th>Max inlet concentration (ppm)</th>
<th>Max outlet concentration (ppm)</th>
<th>Contaminant loading (g contaminant/T batch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T11</td>
<td>U1</td>
<td>500</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>T21</td>
<td>U1</td>
<td>500</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>T12</td>
<td>U2</td>
<td>0</td>
<td>400</td>
<td>100</td>
</tr>
<tr>
<td>T22</td>
<td>U2</td>
<td>0</td>
<td>400</td>
<td>100</td>
</tr>
<tr>
<td>T13</td>
<td>U1</td>
<td>500</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>T23</td>
<td>U1</td>
<td>500</td>
<td>1000</td>
<td>100</td>
</tr>
</tbody>
</table>

4.4.3.1. Results and discussion

The resulting production schedule of Scenario 1 is illustrated in Figure 4.12. The objective achieved was $4642 from the obtained revenue of $7007 and freshwater usage of 3153.2 kg.
The inclusion of sequence dependent changeover constraints in Scenario 2 resulted in the production schedule illustrated in Figure 4.13. Scenario 2 returned revenue of $7155.2, profit of $5527.8 and associated water usage of 2169.8 kg. These represent improvements in profit over Scenario 1 of 19.1% and reduced water usage of 31.2% and were due to the elimination of the washing task between task T22 at p3 and Task T22 at p4 and between T23 at p4 and T23 at p5. Furthermore, the revenue achieved in Scenario 2 increased as a result of an increase in the time available for U2 to process more materials within the time horizon of interest.

Figure 4.14 represents the production schedule of Scenario 4, that is, appropriate task sequencing for water usage with freshwater and direct water reuse constraints. In the figure, the numbers in square brackets represent the amount of water reused directly. Revenue of $7155.2, profit of $5677.8 and water usage of 1969.8 kg were achieved. The production schedules for Scenario 2 (Figure 4.13) and Scenario 4 are almost identical except for the presence of direct water reuse as observed in Figure 4.14. The flexibility added to the model by allowing washing to occur at some time after the completion of a production task, afforded...
Figure 4.14. Scenario 4, case study III

Figure 4.15 represents the production schedule of Scenario 5, that is, freshwater and direct water reuse are available for washing without considering the sequence of tasks. The revenue achieved was $7007, freshwater used was 2630.8 kg and profit was $5033.92. This represents an 8.4% increase in profit and reduced water usage of 16.6% compared to Scenario 1 (freshwater only). These improvements were due to the exploitation of direct water reuse opportunities as observed in Figure 4.15. The results of Scenario 4 however, are superior to Scenario 5 with a 12.7% higher profit and 25.1% water reduction. This further reinforces the point that proper sequencing of tasks can prevent the unnecessary use and generation of freshwater and wastewater associated with washing, respectively, than process integration techniques alone. The computational statistics of the scenarios are presented in Table 4.10.
Table 4.10. Computational statistics of case study III

<table>
<thead>
<tr>
<th>Scenario</th>
<th>P</th>
<th>CPU time (s)</th>
<th>Nodes</th>
<th>RMILP</th>
<th>MILP</th>
<th>B.V.</th>
<th>C.V.</th>
<th>Constraints</th>
<th>Non-zeros</th>
<th>Relative gap %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>0.14</td>
<td>76</td>
<td>9000</td>
<td>4642.2</td>
<td>66</td>
<td>520</td>
<td>1050</td>
<td>2472</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0.50</td>
<td>116</td>
<td>9000</td>
<td>5527.8</td>
<td>176</td>
<td>816</td>
<td>1916</td>
<td>6588</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1.42</td>
<td>132</td>
<td>9000</td>
<td>5677.8</td>
<td>240</td>
<td>1206</td>
<td>2910</td>
<td>9166</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>2.89</td>
<td>1174</td>
<td>9601</td>
<td>5033.9</td>
<td>160</td>
<td>1164</td>
<td>2473</td>
<td>5856</td>
<td>-</td>
</tr>
</tbody>
</table>

4.5. Conclusions

A new mathematical formulation has been developed for the simultaneous optimisation of batch production and wastewater minimisation. The formulation takes advantage of the sequence of tasks in determining both the occurrence and extent of water using operations in a unit. The formulation relies on a flexible schedule featuring continuous time, unit specific event points and was based on fixed mass load applications. Results from the application of the model to two case studies showed that even without process integration, water usage can be minimized through the appropriate selection of task sequences in a unit. Reductions in water usage of 33 % and 48.8 % were observed in case study I and case study II respectively which also improved profits.

The formulation was extended to incorporate process integration in the form of direct water reuse. Application of this model to case study III revealed a further improvement in water use which impacted the profit. Other process integration techniques for water reduction such as water storage and regeneration can also be investigated using this framework.

Nomenclature

Sets

- \( C \) \{ \( c \) \( | \) \( c \) = contaminant \}
- \( J \) \{ \( j \) \( | \) \( j \) = a unit \}
- \( P \) \{ \( p \) \( | \) \( p \) = a time point \}
- \( S \) \{ \( s \) \( | \) \( s \) = any state other than a product \}

© University of Pretoria
Chapter 4. Wastewater Minimisation in Batch Plants with Sequence Dependent Changeover

\[ S_{in,j} \{ s_{in,j} \} \] = tasks performed in unit \( j \)

Continuous variables

- \( c_{out}(s_{in,j}, c, p) \) \hspace{1cm} outlet concentration of contaminant \( c \), from unit \( j \) at time point \( p \)
- \( M(s_{in,j}, c, p) \) \hspace{1cm} mass load of contaminant \( c \) in unit \( j \) at time point \( p \) after processing
- \( m_u(s_{in,j}, p) \) \hspace{1cm} amount of material processed by a task at time point \( p \)
- \( F_{in}(s_{in,j}, p) \) \hspace{1cm} mass of water into unit \( j \) at time point \( p \)
- \( F_{out}(s_{in,j}, p) \) \hspace{1cm} mass of water leaving unit \( j \) at time point \( p \)
- \( F_e(s_{in,j}, p) \) \hspace{1cm} mass of effluent water from unit \( j \) at time point \( p \)
- \( F_f(s_{in,j}, p) \) \hspace{1cm} mass of freshwater into unit \( j \) at time point \( p \)
- \( F_r(s_{in,j}, s_{in,j'}, p) \) \hspace{1cm} mass of water reused from washing unit \( j \) after processing \( s_{in,j} \) to unit \( j' \) after processing \( s_{in,j} \) at time point \( p \)
- \( t_p(s_{in,j}, p) \) \hspace{1cm} time at which a state is produced from unit \( j \) at time point \( p \)
- \( t_u(s_{in,j}, p) \) \hspace{1cm} time at which a state is used in unit \( j \) at time point \( p \)
- \( tw_{in}(s_{in,j}, p) \) \hspace{1cm} time at which washing begins
- \( tw_{out}(s_{in,j}, p) \) \hspace{1cm} time at which washing ends
- \( tw_r(s_{in,j}, s_{in,j'}, p) \) \hspace{1cm} time at which water is reused

Binary variables

- \( x_{ch}(s_{in,j}, s_{in,j'}, p) \) \hspace{1cm} binary variable associated with changeover from \( s_{in,j} \) to \( s_{in,j'} \) at time point \( p \)
- \( XL(s_{in,j}, p) \) \hspace{1cm} binary variable indicating that \( s_{in,j} \) is the last task in the unit at time point \( p \)
- \( y(s_{in,j}, p) \) \hspace{1cm} binary variable associated with usage of state \( s \) in unit \( j \) at time point \( p \)
- \( y_r(s_{in,j}, s_{in,j'}, p) \) \hspace{1cm} binary variable associated with the reuse of water from washing unit \( j \) after processing \( s_{in,j} \) to unit \( j' \) after processing \( s_{in,j} \) at time point \( p \)

© University of Pretoria
Parameters

\( \alpha(s_{in,j}) \)  
constant coefficient of processing time of a task

\( \beta(s_{in,j}) \)  
variable coefficient of processing time of a task

\( C_{in}(s_{in,j}, c) \)  
maximum inlet concentration of contaminant \( c \) in unit \( j \)

\( C_{out}(s_{in,j}, c) \)  
maximum outlet concentration of contaminant \( c \) from unit \( j \)

\( CE \)  
cost of effluent water treatment

\( CF \)  
cost of freshwater

\( CP(s) \)  
cost price of product \( s \)

\( CR(s) \)  
cost price of raw material \( s \)

\( F_{in}(s_{in,j}) \)  
maximum inlet water mass of unit \( j \) after processing \( s_{in,j} \)

\( H \)  
time horizon of interest

\( \Delta M(s_{in,j}, c) \)  
contaminant loading corresponding to processing \( s_{in,j} \)

\( M(s_{in,j}, c) \)  
maximum contaminant loading corresponding to processing \( s_{in,j} \)

\( \tau_{ch}(s_{in,j}, s'_{in,j}) \)  
duration of washing for unit \( j \) when \( s_{in,j} \) is processed before \( s'_{in,j} \)  
(sequence dependent)

\( \tau_{w}(s_{in,j}) \)  
fixed duration of washing for unit \( j \) (sequence independent) after  
performing the task corresponding to \( s_{in,j} \)

References


CHAPTER 5
A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

Introduction

During production, certain tasks may require cooling or heating and washing of equipment may be necessary on completion of the task. These requirements present opportunities for heat integration and wastewater minimisation. The problem addressed in this chapter involves the simultaneous optimisation of water and energy within a production scheduling framework to promote plant profitability.

5.1. Problem statement

The problem can be stated as follows:

Given the following scheduling data:
   i. production recipe for each product,
   ii. available units and their capacities,
   iii. maximum storage capacity for each material,
   iv. task durations,
   v. time horizon of interest or production requirement,
   vi. costs of raw materials,
   vii. selling price of final products,

heat integration data:
   i. hot duties for tasks requiring heating and cold duties for tasks that require cooling,
   ii. operating temperatures of heat sources and heat sinks,
   iii. minimum allowable temperature differences,
   iv. heat capacities of materials,
v. costs of hot and cold utilities,  
vi. design limits on heat storage,  
and wastewater minimisation data  
i. washing time for each unit,  
ii. mass load of contaminants in each unit,  
iii. maximum inlet and outlet concentrations of contaminants for each unit,  
iv. maximum storage available for water reuse,  
v. cost of freshwater and  
vi. cost of effluent treatment

Determine an optimal production schedule that achieves a maximum profit or minimum makespan, requiring the minimum amount of external utilities and freshwater use.

5.2. Mathematical formulation

The mathematical formulation consists of the three modules given in the problem statement, i.e. production scheduling, heat integration and wastewater minimisation modules. At the time this work was published (Adekola, O., Stamp, J., Majozi, T., Garg, A., Bandyopadhyay, S., 2013. Unified approach for the optimisation of energy and water in multipurpose batch plants using a flexible scheduling framework. Ind. Eng. Chem. Res., 52, 8488-8506), the goal was to show the superiority of a simultaneous approach for the optimisation of batch production, energy and water, over a sequential approach in which batch production was performed first followed by energy and water optimisation or vice versa. As such, the scheduling framework of Majozi and Zhu (2001), which features the SSN recipe representation and unit specific continuous time, was adopted. More recent state of the art scheduling formulations have since been developed with important features such as reduced computational time and the effective consideration of FIS. Examples of such works are: Shaik and Floudas (2009), Susarla et al. (2010) and Seid and Majozi (2012).

The necessary scheduling constraints can be found in Majozi and Zhu (2001). The heat integration and wastewater minimisation constraints are based on the superstructure in Figure 5.1 (a) and (b) respectively. The heat integration constraints are presented first, followed by the necessary wastewater minimisation constraints.
The sets, parameters and variables on which the mathematical formulation is based can be found in the nomenclature before the end of this chapter.

Figure 5.1. Superstructure for mathematical formulation: (a) when units perform heating/cooling tasks, (b) when units perform washing tasks.

5.2.1. Heat integration constraints

In Figure 5.1 (a), each processing unit may operate using either direct or indirect heat integration. Processing units 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.
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

Constraints (5.1) to (5.39) constitute the heat integration model, useful for multipurpose batch processes and include direct and indirect heat integration.

Constraints (5.1) and (5.2) 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 the 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.

\[
\sum_{s_{m,j_h}} x(s_{m,j_h} \cdot s_{m,j_a} \cdot p) \leq y(s_{m,j_a} \cdot p) \quad \forall p \in P, \quad s_{m,j_a} \in S_{m,j} \quad (5.1)
\]

\[
\sum_{s_{m,j_a}} x(s_{m,j_a} \cdot s_{m,j_h} \cdot p) \leq y(s_{m,j_a} \cdot p) \quad \forall p \in P, \quad s_{m,j_a} \in S_{m,j} \quad (5.2)
\]

Constraints (5.3) and (5.4) ensure that heat integration between a unit and heat storage may occur only if the unit is active at that time point. However, if a unit is active, it will not necessarily integrate with heat storage.

\[
z(s_{m,j_a} \cdot u, p) \leq y(s_{m,j_a} \cdot p) \quad \forall p \in P, \quad s_{m,j_a} \in S_{m,j}, \quad u \in U \quad (5.3)
\]

\[
z(s_{m,j_h} \cdot u, p) \leq y(s_{m,j_h} \cdot p) \quad \forall p \in P, \quad s_{m,j_h} \in S_{m,j}, \quad u \in U \quad (5.4)
\]

Constraint (5.5) ensures that heat storage is heat integrated with either one hot unit or one cold unit at any point in time. This is to simplify and improve operational efficiency in the plant.

\[
\sum_{s_{m,j_h}} z(s_{m,j_h} \cdot u, p) + \sum_{s_{m,j_a}} z(s_{m,j_a} \cdot u, p) \leq 1, \quad \forall p \in P, \quad u \in U \quad (5.5)
\]

Constraints (5.6) and (5.7) ensure that a unit cannot simultaneously undergo direct and indirect heat integration. This condition simplifies the operation of the process.
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\[
\sum_{s_{m,j}} x(s_{m,j}, s_{m,j}, p) + z(s_{m,j}, u, p) \leq 1, \quad \forall \ p \in P, \ s_{m,j} \in S_{m,j}, \ u \in U (5.6)
\]

\[
\sum_{s_{m,j}} x(s_{m,j}, s_{m,j}, p) + z(s_{m,j}, u, p) \leq 1, \quad \forall \ p \in P, \ s_{m,j} \in S_{m,j}, \ u \in U (5.7)
\]

Constraints (5.8) and (5.9) 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, \( z(s_{m,j}, u, p) \), is zero. These constraints are active over the entire time horizon, where \( p \) is the current time point and \( p - 1 \) is the previous time point.

\[
Q(s_{m,j}, u, p) = W(u)C_{\text{fluid}} (T_i(u, p-1) - T_f(u, p)) z(s_{m,j}, u, p - 1),
\]

\[
\forall \ p \in P, \ p > p_1, \ s_{m,j} \in S_{m,j}, \ u \in U
\]

(5.8)

\[
Q(s_{m,j}, u, p) = W(u)C_{\text{fluid}} (T_f(u, p) - T_i(u, p - 1)) z(s_{m,j}, u, p - 1),
\]

\[
\forall \ p \in P, \ p > p_1, \ s_{m,j} \in S_{m,j}, \ u \in U
\]

(5.9)

Constraint (5.10) quantifies the heat transferred to the heat storage vessel at the beginning of the time horizon. The initial temperature of the heat storage fluid is \( T_i(u, p_1) \).

\[
Q(s_{m,j}, u, p_1) = W(u)C_{\text{fluid}} (T_f(u, p_1) - T_i(u, p_1)) z(s_{m,j}, u, p_1),
\]

\[
\forall \ s_{m,j} \in S_{m,j}, \ u \in U
\]

(5.10)

Constraint (5.11) 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.

\[
T_i(u, p) = T_i(u, p - 1), \quad \forall \ p \in P, \ u \in U
\]

(5.11)
Constraints (5.12) and (5.13) ensure that temperature of heat storage does not change if there is no heat integration with the heat storage unit unless there is heat loss from the heat storage unit. \( MM \) is any large number, thereby resulting in an overall “Big M” formulation. If either \( z(s_{m,j}, u, p) \) or \( z(s_{m,j}, u, p-1) \) is equal to one, Constraint (5.12) and Constraint (5.13) 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.

\[
T_i(u, p) - T_j(u, p) - MM \left( \sum_{s_{in,j}} z(s_{m,j}, u, p) + \sum_{s_{in,j}} z(s_{m,j}, u, p-1) \right) \\
\forall \ p \in P, p > p_1, \ u \in U
\]

(5.12)

\[
T_i(u, p) - T_j(u, p) - MM \left( \sum_{s_{in,j}} z(s_{m,j}, u, p) + \sum_{s_{in,j}} z(s_{m,j}, u, p-1) \right) \\
\forall \ p \in P, p > p_1, \ u \in U
\]

(5.13)

Constraint (5.14) ensures that minimum thermal driving forces are obeyed when there is direct heat integration between a hot and a cold unit. This constraint holds when both hot and cold units operate at constant temperature, which is commonly encountered in practice. An example is when there is heat integration between an exothermic and an endothermic reaction.

\[
T(s_{in,j}) - T(s_{in,j}) \geq \Delta T_{\text{min}} - MM \left( 1 - x(s_{in,j}, s_{in,j}, p-1) \right), \\
\forall \ p \in P, p > p_1, \ s_{in,j}, s_{in,j} \in S_{m,j}
\]

(5.14)

Constraints (5.15) and (5.16) ensure that minimum thermal driving forces are obeyed when there is heat integration with the heat storage unit. Constraint (5.15) applies for heat integration between heat storage and a heat sink while constraint (5.16) applies for heat integration between heat storage and a heat source. In Constraints (5.15) and (5.16), the units operate at fixed temperatures. For units not operating at fixed temperatures, both inlet and outlet minimal thermal driving forces between the two integrated tasks need also to be enforced.
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\[
T_f(u, p) - T(s_{in,j}) \geq \Delta T_{\text{min}} - MM(1 - z(s_{in,j}, u, p - 1)),
\forall \ p \in P, \ p > p_1, \ s_{in,j} \in S_{in,j}, \ u \in U
\]  
(5.15)

\[
T(s_{in,j}) - T_f(u, p) \geq \Delta T_{\text{min}} - MM(1 - z(s_{in,j}, u, p - 1)),
\forall \ p \in P, \ p > p_1, \ s_{in,j} \in S_{in,j}, \ u \in U
\]  
(5.16)

Constraints (5.17) and (5.18) give the heating load for a cold state and cooling load for a hot state, respectively, for variable batch size and changing temperature.

\[
HL(s_{in,j}, p) = B(s_{in,j}, p)C_p \text{state}(s_{in,j})T_{out}(s_{in,j}) - T_{in}(s_{in,j}),
\forall \ p \in P, \ s_{in,j} \in S_{in,j}
\]  
(5.17)

\[
CL(s_{in,j}, p) = B(s_{in,j}, p)C_p \text{state}(s_{in,j})T_{in}(s_{in,j}) - T_{out}(s_{in,j}),
\forall \ p \in P, \ s_{in,j} \in S_{in,j}
\]  
(5.18)

Constraint (5.19) ensures that the heating of a cold state will be satisfied by either direct or indirect heat integration as well as external utility if required.

\[
HL(s_{in,j}, p) = Q(s_{in,j}, u, p) + s_f(s_{in,j}, p) + \sum_{s_{in,j}} q(s_{in,j}, s_{in,j}),
\forall \ p \in P, \ s_{in,j} \in S_{in,j}
\]  
(5.19)

Constraint (5.20) states that the cooling of a hot state will be satisfied by either direct or indirect heat integration as well as external utility if required.

\[
CL(s_{in,j}, p) = Q(s_{in,j}, u, p) + c_w(s_{in,j}, p) + \sum_{s_{in,j}} q(s_{in,j}, s_{in,j}),
\forall \ p \in P, \ s_{in,j} \in S_{in,j}
\]  
(5.20)

The upper bounds of the heating load of a cold state, the cooling load of a hot state and the amount of heat exchanged during direct integration are given in Constraints (5.21) to (5.23).
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\[ HL(s_{in,j}, p) \leq y(s_{in,j}, p)Q_{max}(s_{in,j}) \quad \forall p \in P, \ s_{in,j} \in S_{in,j} \]  
(5.21)

\[ CL(s_{in,j}, p) \leq y(s_{in,j}, p)Q_{max}(s_{in,j}) \quad \forall p \in P, \ s_{in,j} \in S_{in,j} \]  
(5.22)

\[ q(s_{in,j}, s_{in,k}, p) \leq x(s_{in,j}, s_{in,k}, p)\min\{Q_{max}(s_{in,j}), Q_{max}(s_{in,k})\} \quad \forall p \in P, \ s_{in,j}, s_{in,k} \in S_{in,j} \]  
(5.23)

For the specific case where the heating and cooling loads are fixed, Constraints (5.24) and (5.25) are used instead of (5.19) and (5.20).

\[ HL(s_{in,j})y(s_{in,j}, p) = Q(s_{in,j}, u, p) + st(s_{in,j}, p) \]

\[ + x(s_{in,j}, s_{in,j}, p)\sum_{s_{in,j}, s_{in,j}, s_{in,j}} \min \{HL(s_{in,j}), CL(s_{in,j})\} \]

\[ \forall p \in P, \ s_{in,j} \in S_{in,j}, \ u \in U \]  
(5.24)

\[ CL(s_{in,j})y(s_{in,j}, p) = Q(s_{in,j}, u, p) + cw(s_{in,j}, p) \]

\[ + x(s_{in,j}, s_{in,j}, p)\sum_{s_{in,j}, s_{in,j}, s_{in,j}} \min \{HL(s_{in,j}), CL(s_{in,j})\} \]

\[ \forall p \in P, \ s_{in,j} \in S_{in,j}, \ u \in U \]  
(5.25)

The amount of heat transferred through direct heat integration will be limited by the smaller heating or cooling requirement of the heat integrated tasks. Constraints (5.26) and (5.27) express this. Constraint (5.26) calculates the heat load of the cold task while Constraint (5.27) calculates the cooling load of the hot task.

\[ q(s_{in,j}, s_{in,k}, p) \leq B(s_{in,j}, p)C_p(s_{in,j})(T_{out}(s_{in,j}) - T_{in}(s_{in,j}))x(s_{in,j}, s_{in,j}, p) \]

\[ \forall p \in P, \ s_{in,j}, s_{in,k} \in S_{in,j} \]  
(5.26)

\[ q(s_{in,j}, s_{in,k}, p) \leq B(s_{in,j}, p)C_p(s_{in,j})(T_{in}(s_{in,j}) - T_{out}(s_{in,j}))x(s_{in,j}, s_{in,j}, p) \]

\[ \forall p \in P, \ s_{in,j}, s_{in,k} \in S_{in,j} \]  
(5.27)
Furthermore, it is possible that a given pair of tasks cannot be heat integrated or that a possible $\Delta T^{\text{min}}$ violation may occur. In this work, the possibility of heat integration between pairs of tasks as well as possible $\Delta T^{\text{min}}$ violations was investigated for each pair of hot and cold tasks beforehand. If $\Delta T^{\text{min}}$ violations occur, the temperatures in Constraints (5.26) and (5.27) should be adjusted for this.

The amount of heat transferred through direct heat integration can also be limited by the duration of the shorter task if the tasks have different durations. Constraints (5.28) to (5.33) capture this. Constraints (5.28) and (5.29) calculate the heating load per time and cooling load per time, of the cold task and hot task, respectively.

\[
\dot{HL}(s_{m,j}, p) = \frac{HL(s_{m,j}, p)}{\text{dur}(s_{m,j}, p)}, \quad \forall \ p \in P, \ s_{m,j} \in S_{m,j} \tag{5.28}
\]

\[
\dot{CL}(s_{m,j}, p) = \frac{CL(s_{m,j}, p)}{\text{dur}(s_{m,j}, p)}, \quad \forall \ p \in P, \ s_{m,j} \in S_{m,j} \tag{5.29}
\]

Constraint (5.30) calculates the heat load of the cold task based on the duration of the same cold task. Constraint (5.31) calculates the heat load of the cold task based on the duration of the hot task. Constraint (5.32) calculates the cooling load of the hot task based on the duration of the same hot task. Constraint (5.33) calculates the cooling load of the hot task based on the duration of the cold task. The amount of heat integrated directly will effectively be the minimum of these four quantities.

\[
q(s_{m,j}, s_{in,j}, p) \leq \dot{HL}(s_{m,j}, p) \text{dur}(s_{m,j}, p), \quad \forall \ p \in P, \ s_{m,j}, s_{in,j} \in S_{m,j} \tag{5.30}
\]

\[
q(s_{m,j}, s_{in,j}, p) \leq \dot{HL}(s_{m,j}, p) \text{dur}(s_{m,j}, p), \quad \forall \ p \in P, \ s_{m,j}, s_{in,j} \in S_{m,j} \tag{5.31}
\]

\[
q(s_{m,j}, s_{in,j}, p) \leq \dot{CL}(s_{m,j}, p) \text{dur}(s_{m,j}, p), \quad \forall \ p \in P, \ s_{m,j}, s_{in,j} \in S_{m,j} \tag{5.32}
\]

\[
q(s_{m,j}, s_{in,j}, p) \leq \dot{CL}(s_{m,j}, p) \text{dur}(s_{m,j}, p), \quad \forall \ p \in P, \ s_{m,j}, s_{in,j} \in S_{m,j} \tag{5.33}
\]
In Constraints (5.28) to (5.33), the duration is a function of batch size. If the duration is fixed, \( \tau(s_{m,j}) \) is used and these constraints are then linear.

Constraints (5.34) and (5.35) ensure that the times at which units are active are synchronized when direct heat integration takes place. Starting 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.

\[
t_u(s_{m,j}, p) \geq t_u(s_{m,j}, p) - MM \left( 1 - x(s_{m,j}, s_{m,j}, p) \right)
\]
\[\forall p \in P, \ s_{m,j}, s_{m,j} \in S_{m,j} \tag{5.34}\]

\[
t_u(s_{m,j}, p) \leq t_u(s_{m,j}, p) + MM \left( 1 - x(s_{m,j}, s_{m,j}, p) \right)
\]
\[\forall p \in P, \ s_{m,j}, s_{m,j} \in S_{m,j} \tag{5.35}\]

Constraints (5.36) and (5.37) ensure that if indirect heat integration takes place, the time at which a heat storage unit starts either to transfer or receive heat will be equal to the time a unit is active.

\[
t_u(s_{m,j}, p) \geq t_i(s_{m,j}, u, p) - MM \left( y(s_{m,j}, p) - z(s_{m,j}, u, p) \right)
\]
\[\forall p \in P, \ u \in U, \ s_{m,j} \in S_{m,j} \tag{5.36}\]

\[
t_u(s_{m,j}, p) \leq t_i(s_{m,j}, u, p) + MM \left( y(s_{m,j}, p) - z(s_{m,j}, u, p) \right)
\]
\[\forall p \in P, \ u \in U, \ s_{m,j} \in S_{m,j} \tag{5.37}\]

Constraints (5.38) and (5.39) 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.

\[
t_u(s_{m,j}, p - 1) + \tau(s_{m,j})y(s_{m,j}, p - 1) \geq t_f(s_{m,j}, u, p)
\]
\[- MM \left( y(s_{m,j}, p - 1) - z(s_{m,j}, u, p - 1) \right) \]
\[ \forall \ p \in P, \ p > p_1, \ u \in U, \ s_{in,j} \in S_{in,j} \quad (5.38) \]

\[ t_u(s_{in,j}, p - 1) + \tau(s_{in,j})y(s_{in,j}, s_{in,j}, p - 1) \leq t_u(s_{in,j}, u, p) \]

\[ + MM \left( y(s_{in,j}, p - 1) - z(s_{in,j}, u, p - 1) \right) \]

\[ \forall \ p \in P, \ p > p_1, \ u \in U, \ s_{in,j} \in S_{in,j} \quad (5.39) \]

5.2.2. Wastewater minimisation constraints

The wastewater minimisation constraints are based on the superstructure in Figure 5.1 (b). The task being performed in unit \( j \in J_w \) is a washing operation in which the water used could consist of freshwater, stored water or recycled/reused water. Water from unit \( j \) can be recycled into the same unit, reused by other units or sent to storage. Direct water reuse refers to the use of an outlet wastewater stream from one unit in another unit, while indirect water reuse refers to the use of previously stored wastewater in a unit.

Constraints (5.40) to (5.90) constitute the wastewater minimisation model useful for multipurpose batch processes, which involve multiple contaminants. The formulation includes both direct water reuse and indirect water reuse due to the presence of a central storage vessel.

Mass balances around each processing unit and the central storage vessel are formulated as follows.

5.2.2.1. Mass balance around a unit

Constraint (5.40) is the water balance over the inlet to a unit. Water entering the unit is a combination of reuse/recycle streams from other units, \( j' \), freshwater and water from storage. Constraint (5.41) states that the water leaving a unit could be recycled/reused, sent to storage or discarded as effluent. Constraint (5.42) states that the amount of water exiting a unit must equal the amount of water entering the unit at the previous time point. This constraint captures the fact that water is neither produced nor lost in the unit during the washing operation.
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\[ F_{in}(s_{in,j}, p) = \sum_{s_{in,j}'} F_j(s_{in,j}', s_{in,j}, p) + F_j(s_{in,j}, p) + F_{out}(s_{in,j}, p) \]

\[ \forall j, j' \in J_w, s_{in,j} \in S_{in,j}, p \in P \] (5.40)

\[ F_{out}(s_{in,j}, p) = \sum_{s_{in,j}'} F_j(s_{in,j}', s_{in,j}, p) + F_j(s_{in,j}, p) + F_{in}(s_{in,j}, p) \]

\[ \forall j, j' \in J_w, s_{in,j} \in S_{in,j}, p \in P \] (5.41)

\[ F_{in}(s_{in,j}, p - 1) = F_{out}(s_{in,j}, p) \]

\[ \forall j \in J_w, s_{in,j} \in S_{in,j}, p \in P, p > p_1 \] (5.42)

Constraint (5.43) represents the inlet contaminant mass balance. The contaminant mass load in the inlet stream is the sum of the contaminant mass load in recycle/reuse water and that in water from storage. Constraint (5.44) represents the outlet contaminant mass as the sum of the mass of contaminant that entered the unit at the previous time point and the mass load of contaminant picked up in the unit during washing. In Constraint (5.44), the mass load of contaminant in a unit is a given parameter.

\[ F_{in}(s_{in,j}, p) c_{in}(s_{in,j}, c, p) = \sum_{s_{in,j}'} F_j(s_{in,j}', s_{in,j}, p) c_{out}(s_{in,j}', c, p) \]

\[ + F_{out}(s_{in,j}, p) c_{out}(c, p) \]

\[ \forall j, j' \in J_w, s_{in,j} \in S_{in,j}, p \in P, c \in C \] (5.43)

\[ F_{out}(s_{in,j}, p) c_{out}(s_{in,j}, c, p) = M(s_{in,j}, c) y_{w}(s_{in,j}, p - 1) \]

\[ + F_{in}(s_{in,j}, p - 1) c_{in}(s_{in,j}, c, p - 1) \]

\[ \forall j \in J_w, s_{in,j} \in S_{in,j}, p \in P, p > p_1, c \in C \] (5.44)

In the case where the mass load of contaminant in a unit is defined as a function of batch size of material processed in the unit, Constraint (5.45) represents the outlet contaminant mass. Constraint (5.46) defines the variable contaminant mass load as the product of the contaminant loading and the batch size of material processed.
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\[ F_{out}(s_{in,j}, p)c_{out}(s_{in,j}, c, p) = M_B(s_{in,j}, c, p - 1)y_w(s_{in,j}, p - 1) \]
\[ + F_{in}(s_{in,j}, p - 1)c_{in}(s_{in,j}, c, p - 1) \]
\[ \forall \ j \in J_w, \ s_{in,j} \in S_{in,j}, \ p \in P, \ p > p_t, \ c \in C \]  \hspace{1cm} (5.45)

\[ M_B(s_{in,j}, c, p) = \Delta M_B(s_{in,j}, c)B(s_{in,j}, p) \]
\[ \forall \ j \in J_w, \ s_{in,j} \in S_{in,j}, \ p \in P, \ c \in C \]  \hspace{1cm} (5.46)

Constraints (5.47) and (5.48) ensure that the inlet and outlet contaminant concentrations do not exceed the allowed maximum. Similarly, the maximum allowable water in a unit must not be exceeded. This is governed by Constraint (5.49). Constraint (5.50) restricts the mass of water entering the unit from recycle/reuse, to the maximum allowable for the unit. Likewise, Constraint (5.51) restricts the mass of water entering the unit from storage, to the maximum allowable for the unit.

\[ c_{in}(s_{in,j}, c, p) \leq C_{in}^U(s_{in,j}, c)y_w(s_{in,j}, p) \]
\[ \forall \ j \in J_w, \ s_{in,j} \in S_{in,j}, \ p \in P, \ c \in C \]  \hspace{1cm} (5.47)

\[ c_{out}(s_{in,j}, c, p) \leq C_{out}^U(s_{in,j}, c)y_w(s_{in,j}, p - 1) \]
\[ \forall \ j \in J_w, \ s_{in,j} \in S_{in,j}, \ p \in P, \ p > p_t, \ c \in C \]  \hspace{1cm} (5.48)

\[ F_{in}(s_{in,j}, p) \leq F_U(s_{in,j})y_w(s_{in,j}, p) \]
\[ \forall \ j \in J_w, \ s_{in,j} \in S_{in,j}, \ p \in P \]  \hspace{1cm} (5.49)

\[ F_r(s_{in,j}, s_{in,j}, p) \leq F_U(s_{in,j})y_w(s_{in,j}, s_{in,j}, p) \]
\[ \forall \ j, j' \in J_w, \ s_{in,j} \in S_{in,j}, \ p \in P \]  \hspace{1cm} (5.50)

\[ F_{s_{out}}(s_{in,j}, p) \leq F_U(s_{in,j})y_{s_{out}}(s_{in,j}, p) \]
\[ \forall \ j \in J_w, \ s_{in,j} \in S_{in,j}, \ p \in P \]  \hspace{1cm} (5.51)
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

The maximum quantity of water into a unit is calculated using Equation (5.52). It is important to note that for multi-contaminant wastewater the outlet concentration of the individual components cannot all be set to the maximum since the contaminants are not limiting simultaneously. The limiting contaminant(s) will always be at the maximum outlet concentration and the non-limiting contaminants will be below their respective maximum outlet concentrations.

\[
F^U(s_{\text{in},j}) = \max_{c \in C} \left\{ \frac{M(s_{\text{in},j}, c)}{C_{\text{out}}(s_{\text{in},j}, c) - C_{\text{in}}(s_{\text{in},j}, c)} \right\}
\]

\forall \ j \in J_w, \ s_{\text{in},j} \in S_{\text{in},j}, \ c \in C \tag{5.52}

5.5.2.2. Mass balance around central storage

Constraint (5.53) is the water mass balance around the storage tank. The amount of water stored in the storage tank consists of water stored at the previous time point and the difference between water entering the storage tank from a unit and water leaving the storage tank to a unit. Constraint (5.54) defines the initial amount of water in the tank.

\[
q_w(p) = q_w(p-1) + \sum_{s_{\text{in},j}} F_{\text{in}}(s_{\text{in},j}, p) - \sum_{s_{\text{out},j}} F_{\text{out}}(s_{\text{in},j}, p)
\]

\forall \ p \in P, \ p > p_1 \tag{5.53}

\[
q_w(p_1) = Q_w - \sum_{s_{\text{out},j}} F_{\text{out}}(s_{\text{in},j}, p_1) \tag{5.54}
\]

The definition of the inlet contaminant concentration to the storage tank is given in Constraint (5.55). The concentration of water exiting the storage tank is assumed to be equal to the concentration of water in the tank as given in Constraint (5.56). This condition is true in the case of perfect mixing. The initial concentration in the storage tank is expressed in Constraint (5.57). Constraint (5.58) ensures that the maximum capacity of the tank is not exceeded.

© University of Pretoria
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\[
cs_{in}(c, p) = \frac{\sum_{s_{in,j}} (Fs_{in}(s_{in,j}, p) c_{out}(s_{in,j}, c, p))}{\sum_{s_{in,j}} Fs_{in}(s_{in,j}, p)} \\
\forall \ s_{in,j} \in S_{in,j}, \ p \in P, \ c \in C
\]

\[
(5.55)
\]

\[
cs_{out}(c, p) = \frac{qw_{i}(p-1)cs_{out}(c, p-1) + \left[ \sum_{s_{in,j}} Fs_{in}(s_{in,j}, p) \right] cs_{in}(c, p)}{qw_{i}(p-1) + \sum_{s_{in,j}} Fs_{in}(s_{in,j}, p)} \\
\forall \ p \in P, \ p \geq p_{i}, \ c \in C
\]

\[
(5.56)
\]

\[
cs_{out}(c, p_{i}) = CS^{o}_{out}(c) \quad \forall \ c \in C
\]

\[
(5.57)
\]

\[
qw_{i}(p) \leq Qw^{U}_{i} \quad \forall \ p \in P
\]

\[
(5.58)
\]

Constraint (5.59) ensures that no water is stored in the storage vessel at the end of the time horizon in order to give a true optimum. Otherwise the resulting minimum effluent could be misleading.

\[
qw_{i}(p) = 0 \quad \forall \ p = |P|
\]

\[
(5.59)
\]

The scheduling constraints for the wastewater minimisation model are as follows.

5.2.2.3. Task scheduling constraints

Unlike heating or cooling that can occur during a production task, a unit can only perform either a production task or a washing task at any given point in time. The task scheduling constraints ensure the proper sequencing of equipment washing and production tasks. Constraint (5.60) expresses the duration of the production task (Majozi and Zhu, 2001). Constraint (5.61) stipulates that washing can only commence at time point \( p \) if the unit was active at the previous time point performing a production task, i.e., \( y(s_{in,j}, p-1) \) has a value of 1. Constraints (5.62) and (5.63) together ensure that unit \( j \) is washed immediately after a
production task \( s_{i,n,j} \), has ended in the unit. If washing is being performed in the unit, \( y_{w}(s_{i,n,j}, p) \) has a value of 1 causing Constraints (5.62) and (5.63) to become active and the start time of washing is forced to coincide with the end time of production. Otherwise, when water is not used in the unit, i.e., \( y_{w}(s_{i,n,j}, p) \) has a value of zero, the two constraints become relaxed. Constraint (5.64) represents the duration of the washing task performed in unit \( j \).

\[
t_{out}(s_{i,n,j}, p) = t_{w}(s_{i,n,j}, p - 1) + \tau(s_{i,n,j})y(s_{i,n,j}, p - 1)
\]

\[\forall j \in J_w, \ s_{i,n,j} \in S_{i,n,j}, \ p \in P, \ p > p_i \]  \hspace{1cm} (5.60)

\[
y_{w}(s_{i,n,j}, p) = y(s_{i,n,j}, p - 1) \quad \forall j \in J_w, \ s_{i,n,j} \in S_{i,n,j}, \ p \in P, \ p > p_i \]  \hspace{1cm} (5.61)

\[
t_{w}(s_{i,n,j}, p) \geq t_{out}(s_{i,n,j}, p) - MM(1 - y_{w}(s_{i,n,j}, p))
\]

\[\forall j \in J_w, \ s_{i,n,j} \in S_{i,n,j}, \ p \in P \]  \hspace{1cm} (5.62)

\[
t_{w}(s_{i,n,j}, p) \leq t_{out}(s_{i,n,j}, p) + MM(1 - y_{w}(s_{i,n,j}, p))
\]

\[\forall j \in J_w, \ s_{i,n,j} \in S_{i,n,j}, \ p \in P \]  \hspace{1cm} (5.63)

\[
t_{w out}(s_{i,n,j}, p) = t_{w}(s_{i,n,j}, p - 1) + \tau_{w}(s_{i,n,j})y_{w}(s_{i,n,j}, p - 1)
\]

\[\forall j \in J_w, \ s_{i,n,j} \in S_{i,n,j}, \ p \in P, \ p > p_i \]  \hspace{1cm} (5.64)

5.2.2.4. Recycle/reuse scheduling constraints

Wastewater can only be recycled/reused directly, if the unit producing wastewater and the unit receiving wastewater finish operating and begin operating at the same time, respectively. Constraint (5.65) describes the relationship between usage of water in a unit and the opportunity for recycle and reuse. The constraint states that for a unit \( j \) to transfer water to unit \( j' \), unit \( j' \) should require water at that time point. It does not, however, mean that unit \( j' \) must use water from unit \( j \), it could still obtain water from other sources. Constraints (5.66) and (5.67) state that the time at which water recycle/reuse takes place coincides with the time at which the water is produced. Constraints (5.68) and (5.69) ensure that the time at
which water recycle/ reuse takes place coincides with the starting time of the unit receiving the water.

\[ y_{w, m, j}(s_{in, j}, s_{in, j'}, p) \leq y_{w}(s_{in, j}, p) \]
\[ \forall j, j' \in J_w, \ s_{in, j} \in S_{in, j}, \ p \in P \]  

(5.65)

\[ tw_r(s_{in, j}, s_{in, j'}, p) \leq tw_{out}(s_{in, j}, p) - MM(1 - y_{w, m, j}(s_{in, j}, s_{in, j'}, p)) \]
\[ \forall j, j' \in J_w, \ s_{in, j} \in S_{in, j}, \ p \in P \]  

(5.66)

\[ tw_r(s_{in, j}, s_{in, j'}, p) \geq tw_{out}(s_{in, j}, p) + MM(1 - y_{w, m, j}(s_{in, j}, s_{in, j'}, p)) \]
\[ \forall j, j' \in J_w, \ s_{in, j} \in S_{in, j}, \ p \in P \]  

(5.67)

\[ tw_r(s_{in, j}, s_{in, j'}, p) \leq tw_{in}(s_{in, j'}, p) - MM(1 - y_{w, m, j}(s_{in, j}, s_{in, j'}, p)) \]
\[ \forall j, j' \in J_w, \ s_{in, j} \in S_{in, j}, \ p \in P \]  

(5.68)

\[ tw_r(s_{in, j}, s_{in, j'}, p) \geq tw_{in}(s_{in, j'}, p) + MM(1 - y_{w, m, j}(s_{in, j}, s_{in, j'}, p)) \]
\[ \forall j, j' \in J_w, \ s_{in, j} \in S_{in, j}, \ p \in P \]  

(5.69)

5.2.2.5 Central storage scheduling constraints

Constraint (5.70) relates water usage in a unit and water transfer from storage. It states that water can only be transferred to a unit if it uses water at the same time point. However, it is not a prerequisite for the unit to use stored water, the water could be provided from other sources. Constraints (5.71) and (5.72) ensure that the time at which water is sent from storage to a unit coincides with the start time of washing of the unit.

\[ y_{s, m, j}(s_{in, j}, p) \leq y_{w}(s_{in, j}, p) \]
\[ \forall j \in J_w, \ s_{in, j} \in S_{in, j}, \ p \in P \]  

(5.70)

\[ ts_{out}(s_{in, j}, p) \geq tw_{in}(s_{in, j}, p) - MM(2 - y_{s, m, j}(s_{in, j}, p) - y_{w}(s_{in, j}, p)) \]
\[ \forall j \in J_w, \ s_{in,j} \in S_{in,j}, \ p \in P \] (5.71)

\[ ts_{out}(s_{in,j}, p) \leq tw_{in}(s_{in,j}, p) + MM(2 - y_{s_{out}}(s_{in,j}, p) - yw(s_{in,j}, p)) \]
\[ \forall j \in J_w, s_{in,j} \in S_{in,j}, p \in P \] (5.72)

Constraint (5.73) relates water usage in a unit and water transfer to storage. It states that water can only be transferred from a unit to storage if the unit used water at the previous time point. However, washing can take place in the unit without discharging water to the storage tank. The water could be discharged to other sinks. Constraints (5.74) and (5.75) ensure that the time at which water is sent to storage from a unit must coincide with the finishing time of washing of the unit.

\[ y_{s_{in}}(s_{in,j}, p) \leq yw(s_{in,j}, p - 1) \quad \forall j \in J_w, s_{in,j} \in S_{in,j}, p \in P, p > p_1 \] (5.73)

\[ ts_{in}(s_{in,j}, p) \geq tw_{out}(s_{in,j}, p) - MM(2 - y_{s_{in}}(s_{in,j}, p) - yw(s_{in,j}, p - 1)) \]
\[ \forall j \in J_w, s_{in,j} \in S_{in,j}, p \in P, p > p_1 \] (5.74)

\[ ts_{in}(s_{in,j}, p) \leq tw_{out}(s_{in,j}, p) + MM(2 - y_{s_{in}}(s_{in,j}, p) - yw(s_{in,j}, p - 1)) \]
\[ \forall j \in J_w, s_{in,j} \in S_{in,j}, p \in P, p > p_1 \] (5.75)

If water is transferred from storage to a unit at a later time point, the time at which this happens must correspond to a later time in the time horizon. This is specified in Constraint (5.76). Constraint (5.77) ensures that if water is transferred from a unit to storage at a later time point, the time at which this happens corresponds to a later time in the time horizon.

\[ ts_{out}(s_{in,j}, p) \geq ts_{out}(s_{in,j}, p') - MM(2 - y_{s_{out}}(s_{in,j}, p) - y_{s_{out}}(s_{in,j}, p')) \]
\[ \forall j, j' \in J_w, s_{in,j} \in S_{in,j}, p, p' \in P, p \geq p' \] (5.76)

\[ ts_{in}(s_{in,j}, p) \geq ts_{in}(s_{in,j}, p') - MM(2 - y_{s_{in}}(s_{in,j}, p) - y_{s_{in}}(s_{in,j}, p')) \]
\[ \forall j, j' \in J_w, s_{in,j} \in S_{in,j}, p, p' \in P, p \geq p' \] (5.77)
Constraints (5.78) and (5.79) state that if water is transferred to storage from more than one unit at the same time point, the time at which they do so must coincide. Constraints (5.80) and (5.81) state that if water is discharged from storage to more than one unit at the same time point, the time at which the water is discharged must coincide.

\[
\begin{align*}
\text{ts}_{in}(s_{in,j}, p) & \leq \text{ts}_{in}(s_{in,j'}, p) - MM(2 - \text{ys}_{in}(s_{in,j}, p) - \text{ys}_{in}(s_{in,j'}, p)) \\
\forall j, j' \in J_w, \ s_{in,j} & \in S_{in,j}, \ p \in P & (5.78) \\
\text{ts}_{in}(s_{in,j}, p) & \leq \text{ts}_{in}(s_{in,j'}, p) + MM(2 - \text{ys}_{in}(s_{in,j}, p) - \text{ys}_{in}(s_{in,j'}, p)) \\
\forall j, j' \in J_w, \ s_{in,j} & \in S_{in,j}, \ p \in P & (5.79) \\
\text{ts}_{out}(s_{out,j}, p) & \geq \text{ts}_{out}(s_{out,j'}, p) - MM(2 - \text{ys}_{out}(s_{out,j}, p) - \text{ys}_{out}(s_{out,j'}, p)) \\
\forall j, j' \in J_w, \ s_{out,j} & \in S_{out,j}, \ p \in P & (5.80) \\
\text{ts}_{out}(s_{out,j}, p) & \leq \text{ts}_{out}(s_{out,j'}, p) + MM(2 - \text{ys}_{out}(s_{out,j}, p) - \text{ys}_{out}(s_{out,j'}, p)) \\
\forall j, j' \in J_w, \ s_{out,j} & \in S_{out,j}, \ p \in P & (5.81)
\end{align*}
\]

If water is simultaneously being transferred to and discharged from storage, the time at which this happens should coincide. This is given in Constraints (5.82) and (5.83)

\[
\begin{align*}
\text{ts}_{in}(s_{in,j}, p) & \geq \text{ts}_{out}(s_{out,j'}, p) - MM(2 - \text{ys}_{in}(s_{in,j}, p) - \text{ys}_{out}(s_{out,j'}, p)) \\
\forall j, j' \in J_w, \ s_{in,j} & \in S_{in,j}, \ p \in P & (5.82) \\
\text{ts}_{in}(s_{in,j}, p) & \leq \text{ts}_{out}(s_{out,j'}, p) + MM(2 - \text{ys}_{in}(s_{in,j}, p) - \text{ys}_{out}(s_{out,j'}, p)) \\
\forall j, j' \in J_w, \ s_{in,j} & \in S_{in,j}, \ p \in P & (5.83)
\end{align*}
\]

Constraint (5.84) ensures that if water leaves storage at a later time point compared to water entering the storage, the time at which water leaves the storage must correspond to a later time in the time horizon.
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\[ ts_{out}(s_{in,j}, p) \geq ts_{in}(s_{m,j}', p') - MM(2 - ys_{out}(s_{in,j}, p) - ys_{in}(s_{m,j}', p')) \]
\[ \forall j, j' \in J_w, s_{in,j} \in S_{m,j}, p, p' \in P, p \geq p' \]  \hspace{1cm} (5.84)

The following feasibility and time horizon constraints also hold. Constraint (5.85) ensures that if a processing unit \( j \) is reusing water from unit \( j' \) at time point \( p \), then unit \( j' \) cannot reuse water from unit \( j \) at the same time point.

\[ yw_r(s_{in,j}, s_{in,j}', p) + yw_r(s_{in,j}', s_{in,j}, p) \leq 1 \]
\[ \forall j, j' \in J_w, s_{in,j} \in S_{m,j}, p \in P \]  \hspace{1cm} (5.85)

Constraints (5.86) to (5.90) ensure that each event occurs within the time horizon of interest.

\[ tw_{in}(s_{in,j}, p) \leq H \quad \forall j \in J_w, s_{in,j} \in S_{m,j}, p \in P \]  \hspace{1cm} (5.86)
\[ tw_{out}(s_{in,j}, p) \leq H \quad \forall j \in J_w, s_{in,j} \in S_{m,j}, p \in P \]  \hspace{1cm} (5.87)
\[ tw_r(s_{in,j}, s_{in,j}', p) \leq H \quad \forall j, j' \in J_w, s_{in,j} \in S_{m,j}, p \in P \]  \hspace{1cm} (5.88)
\[ ts_{in}(s_{in,j}, p) \leq H \quad \forall j \in J_w, s_{in,j} \in S_{m,j}, p \in P \]  \hspace{1cm} (5.89)
\[ ts_{out}(s_{in,j}, p) \leq H \quad \forall j \in J_w, s_{in,j} \in S_{m,j}, p \in P \]  \hspace{1cm} (5.90)

The objective of the formulation is either the maximisation of profit or the minimisation of makespan. Constraint (5.91) expresses the profit as the difference between the product revenue and the sum of freshwater, effluent treatment, cooling water and steam costs. When the objective is the maximisation of profit, Constraint (5.91) is maximised. As a result, the amount of external utilities, as well as freshwater consumption, is minimised.

\[ \text{Profit} = \sum_{s} \sum_{p} CP(s) d(s, p) - CF \sum_{s_{in,j}} \sum_{p} F_f(s_{in,j}, p) - CE \sum_{s_{out,j}} \sum_{p} F_e(s_{out,j}, p) \]
\[ - \text{Cost}_{cw} \sum_{s_{in,jk}} \sum_{p} cw(s_{in,jk}, p) - \text{Cost}_{st} \sum_{s_{in,jc}} \sum_{p} st(s_{in,jc}, p) \] \hspace{1cm} (5.91)
Constraint (5.92) is the objective function for makespan minimisation. In Constraint (5.92), the quotient of profit and time horizon is maximised. As a result, the makespan is minimised and the amount of external utilities, as well as freshwater consumed, are also minimised using the same reasoning as above.

\[
\max \frac{\text{Profit}}{H} \quad (5.92)
\]

5.3. Solution procedure

The overall model, whether for a profit maximisation problem or a makespan minimisation problem is an MINLP. When solving a profit maximisation problem, the model is linearised and solved as an MILP, the solution of which is then used as a starting point for the exact MINLP model. If the solutions from the two models are equal, the solution is globally optimal, as global optimality can be proven for MILP problems. If the solutions differ, the MINLP solution is locally optimal. The possibility also exists that no feasible starting point is found (Gouws et al., 2008; Adekola and Majozi, 2011 and Stamp and Majozi, 2011).

When solving a makespan minimisation problem, the MINLP cannot be linearised completely to an MILP due to the nonlinear objective function, Constraint (5.92). However, the MINLP was linearised to a relaxed MINLP problem which provides a starting point for the exact MINLP problem. However, the resulting solution to the exact MINLP cannot be guaranteed to be a global optimum.

Constraints (5.8), (5.9) and (5.10) have trilinear terms where a binary variable and two continuous variables are multiplied. Constraints (5.28) to (5.33), (5.43), (5.44), (5.45), (5.55) and (5.56) have bilinear terms where two continuous variables are multiplied. This results in a nonconvex MINLP formulation. The bilinearity resulting from the multiplication of a continuous variable with a binary variable as found in Constraints (5.26) and (5.27) may be handled effectively with the Glover (1975) transformation. This is an exact linearisation technique and as such will not compromise the accuracy of the model. The procedure is demonstrated for Constraint (5.9).
Let
\[ T_j(u, p)z(s_{m,j}, u, p - 1) = \Gamma_1(s_{m,j}, u, p) \] (5.93)

With lower and upper temperature bounds known
\[ T^L \leq T_j(u, p) \leq T^U \] (5.94)

Then
\[ \Gamma_1(s_{m,j}, u, p) \geq T_j(u, p) - T^U (1 - z(s_{m,j}, u, p - 1)) \] (5.95)
\[ \Gamma_1(s_{m,j}, u, p) \leq T_j(u, p) + T^L (1 - z(s_{m,j}, u, p - 1)) \] (5.96)
\[ \Gamma_1(s_{m,j}, u, p) \geq z(s_{m,j}, u, p - 1)T^L \] (5.97)
\[ \Gamma_1(s_{m,j}, u, p) \leq z(s_{m,j}, u, p - 1)T^U \] (5.98)

The result from the Glover transformation for Constraint (5.9) is seen in Constraint (5.99) and includes the addition of one new continuous variable and four new continuous constraints.

\[ Q(s_{m,j}, u, p - 1) = W(u) c_p (\Gamma_1(s_{m,j}, u, p) - \Gamma_2(s_{m,j}, u, p - 1)), \]
\[ \forall p \in P, p \geq p_1, s_{m,j} \in S_{m,j}, u \in U \]
(5.99)

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-Linearisation technique (Sherali and Alameddine, 1992) as discussed by Quesada and Grossmann (1995). This is demonstrated for Constraint (5.99), resulting in Constraints (5.100) to (5.107).

Let
\[ W(u)\Gamma_1(s_{m,j}, u, p) = \Psi_1(s_{m,j}, u, p) \] (5.100)
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

With lower and upper heat storage capacity and temperature bounds known

\[ W^L \leq W(u) \leq W^U \]  
\[ T^L \leq \Gamma_1(s_{m,j}, u, p) \leq T^U \]  

Then

\[ \Psi_1(s_{m,j}, u, p) \geq W^L \Gamma_1(s_{m,j}, u, p) + T^L W(u) - W^L T^L \]  
\[ \Psi_1(s_{m,j}, u, p) \geq W^U \Gamma_1(s_{m,j}, u, p) + T^U W(u) - W^U T^U \]  
\[ \Psi_1(s_{m,j}, u, p) \leq W^U \Gamma_1(s_{m,j}, u, p) + T^U W(u) - W^U T^L \]  
\[ \Psi_1(s_{m,j}, u, p) \leq W^L \Gamma_1(s_{m,j}, u, p) + T^L W(u) - W^L T^U \]  

This is an inexact linearisation technique and increases the size of the model by an additional type of continuous variable and four types of continuous constraints. The final completely linearised form of Constraint (5.9) can be seen in Constraint (5.107).

\[ Q(s_{m,j}, u, p - 1) = c_p (\Psi_1(s_{m,j}, u, p) - \Psi_1(s_{m,j}, u, p - 1)) \]  
\[ \forall \ p \in P, p > p_i, \ s_{m,j} \in S_{m,j}, \ u \in U \]  

(5.107)

For the profit maximisation problem, the linearisation procedure is carried out for each of the nonlinear terms in Constraints (5.8), (5.9), (5.10), (5.26) to (5.33), (5.43), (5.44), (5.45), (5.55) and (5.56) to produce a MILP problem, which provides a starting point for the exact MINLP problem. The objective function for the makespan minimisation problem was also nonlinear, however, only Constraints (5.8), (5.9), (5.10), (5.26) to (5.33), (5.43), (5.44), (5.45), (55) and (56) were linearised to produce a relaxed MINLP problem, the solution of which provides a starting point for the exact MINLP problem.
5.4. Case study I (Halim and Sinivasan, 2011)

This case study is a simple sequential process. A single product is produced via three tasks in five units. The case study has been adapted to include water and energy using operations. Figure 5.2 shows the recipe representation of the process. Steam is available for heating, cooling water is available for cooling and freshwater is available for washing. The objective is to maximise profit over a 12 h time horizon, while minimising freshwater usage and utility requirements.

![Recipe representation of the simple sequential process for case study I](image)

Data required for case study I may be obtained from Table 5.1 to Table 5.3.

**Table 5.1. Data pertaining to production, for case study I**

<table>
<thead>
<tr>
<th>Task (i)</th>
<th>Unit (j)</th>
<th>Max batch size (kg)</th>
<th>Processing time (h)</th>
<th>Washing time (h)</th>
<th>Material state (m)</th>
<th>Initial inventory (kg)</th>
<th>Max storage (kg)</th>
<th>Revenue/cost ($/kg or $/MJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>Unit 1</td>
<td>100</td>
<td>1.25</td>
<td>0.25</td>
<td>A</td>
<td>1000</td>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Unit 2</td>
<td>150</td>
<td>1.70</td>
<td>0.30</td>
<td>B</td>
<td>0</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>Task 2</td>
<td>Unit 3</td>
<td>200</td>
<td>1.50</td>
<td>0</td>
<td>C</td>
<td>0</td>
<td>250</td>
<td>0</td>
</tr>
<tr>
<td>Task 3</td>
<td>Unit 4</td>
<td>100</td>
<td>0.75</td>
<td>0.25</td>
<td>D</td>
<td>0</td>
<td>1000</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Unit 5</td>
<td>150</td>
<td>1.20</td>
<td>0.30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Washwater</td>
<td></td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Wastewater</td>
<td></td>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cooling water</td>
<td></td>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Steam</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

**Table 5.2. Data pertaining to energy requirements, for case study I**

<table>
<thead>
<tr>
<th>Task (i)</th>
<th>$T_a$ (°C)</th>
<th>$T_{out}$ (°C)</th>
<th>Unit (j)</th>
<th>$C_p$ (kJ/kg°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>140</td>
<td>60</td>
<td>Unit 1</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Unit 2</td>
<td>4.0</td>
</tr>
<tr>
<td>Task 2</td>
<td>60</td>
<td>40</td>
<td>Unit 3</td>
<td>3.5</td>
</tr>
<tr>
<td>Task 3</td>
<td>40</td>
<td>80</td>
<td>Unit 4</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Unit 5</td>
<td>3.0</td>
</tr>
<tr>
<td>Cooling water</td>
<td>20</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steam</td>
<td>170</td>
<td>160</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5.3. Data pertaining to water requirements, for case study I

<table>
<thead>
<tr>
<th>Task (i)</th>
<th>Unit (j)</th>
<th>Max inlet concentration (ppm)</th>
<th>Max outlet concentration (ppm)</th>
<th>Contaminant loading (g contaminant/kg batch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>Unit 1</td>
<td>500</td>
<td>1000</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>Unit 2</td>
<td>50</td>
<td>100</td>
<td>0.2</td>
</tr>
<tr>
<td>Task 2</td>
<td>Unit 3</td>
<td>-</td>
<td>-</td>
<td>0.2</td>
</tr>
<tr>
<td>Task 3</td>
<td>Unit 4</td>
<td>150</td>
<td>300</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>Unit 5</td>
<td>300</td>
<td>2000</td>
<td>0.2</td>
</tr>
</tbody>
</table>

The following should be noted:
1. The processing duration of a batch is fixed.
2. The mass load of contaminants is dependent on the batch size.
3. A single contaminant is present
4. The energy requirements of the operations are a function of the batch size and the given initial and final temperatures. Due to this fact, careful consideration must be given to ensure that whenever heat integration occurs between two operations, the minimum temperature difference for heat transfer, $10^\circ C$ is not violated.

The first case study was solved with the proposed formulation, using Constraints (5.1), (5.2), (5.17) to (5.23), (5.26) to (5.35), (5.40) to (5.43) and (5.45) to (5.50), (5.52), (5.60) to (5.69) and (5.85) to (5.88). The objective function was the maximisation of Constraint (5.91). It is important to note that in Constraints (5.40), (5.41), (5.43) and (5.45), variables associated with wastewater storage are not included. Due to the fact that a single contaminant was present, the formulation for this case study could be reduced to a MILP. The outlet contaminant concentration was fixed to the maximum, Constraint (5.43) was substituted into Constraint (5.45) and a Glover transformation was performed on the resulting equation (Majozi, 2005). The computer used to solve the model had an Intel(R) Core(TM) i7-2670QM, 2.2 GHz processor with 4.0GB RAM. The problem was solved with GAMS using CPLEX as the MIP solver.

The results from the first case study as compared to the results achieved by Halim and Srinivasan (2011) may be obtained from Table 5.4.
Table 5.4. Comparison of results for case study I

<table>
<thead>
<tr>
<th></th>
<th>Halim &amp; Srinivasan(2011)</th>
<th>This formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profit ($)</td>
<td>4 764.1</td>
<td>4 775.28</td>
</tr>
<tr>
<td>Steam (MJ)</td>
<td>43.9</td>
<td>66</td>
</tr>
<tr>
<td>Cooling water (MJ)</td>
<td>313.9</td>
<td>336</td>
</tr>
<tr>
<td>Total freshwater (kg)</td>
<td>1 238.37</td>
<td>1 013.33</td>
</tr>
<tr>
<td>Revenue from product ($)</td>
<td>5 000</td>
<td>5 000</td>
</tr>
<tr>
<td>Cost of steam ($)</td>
<td>43.9</td>
<td>66</td>
</tr>
<tr>
<td>Cost of cooling water ($)</td>
<td>6.28</td>
<td>6.72</td>
</tr>
<tr>
<td>Cost of freshwater ($)</td>
<td>123.8</td>
<td>101.3</td>
</tr>
<tr>
<td>Cost of wastewater ($)</td>
<td>61.9</td>
<td>50.7</td>
</tr>
<tr>
<td>Number of slots</td>
<td>7</td>
<td>N/A</td>
</tr>
<tr>
<td>Number of time points</td>
<td>N/A</td>
<td>15</td>
</tr>
<tr>
<td>Number of binary variables</td>
<td>not reported</td>
<td>385</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>1 500</td>
<td>N/A</td>
</tr>
<tr>
<td>CPU time (s)</td>
<td>not reported</td>
<td>28 797</td>
</tr>
</tbody>
</table>

As can be observed from Table 5.4, a better profit overall was obtained with the proposed formulation. Although the amounts of steam and cooling water were lower for the model by Halim and Srinivasan (2011), the amount of freshwater required was lower in the proposed formulation, which also results in a lower effluent production. This is a consequence of differences in the schedule obtained by Halim and Srinivasan (2011) compared to the current formulation. The advantage of the current work is that the variable nature of time ensures that the schedule is flexible and results that could have been overlooked when time is fixed are available during the optimisation search. The CPU time required to solve the problem was approximately 8 h. This highlights the complexity introduced by solving scheduling, wastewater minimisation and heat integration simultaneously, whereas the standalone scheduling problem resulted in an objective of $ 5000 obtained in 0.097 s. Similar to the method by Halim and Srinivasan (2011), the washing of a unit may not necessarily occur immediately after the end of a processing task, but can be delayed to improve reuse opportunities with other units. To cater for this, Constraint (5.63) was removed from the formulation while Constraint (5.62) remained.

Figure 5.3 shows the Gantt chart with the resulting production schedule for the first case study. The clear horizontal blocks represent the tasks associated with production processes. The amount of material processed in each unit is labelled within the clear blocks. The dark horizontal blocks represent the washing of a unit. The amount of water associated with
washing is labelled underneath the relevant dark blocks. The double sided arrows (up-down arrows) represent direct heat integration between two tasks. The numbers in round brackets represent the amount of energy associated with heat integration. The numbers in curly brackets represent steam duties while the numbers in square brackets represent cooling water duties.

Figure 5.3. Resulting production schedule for case study I

5.5. Case study II

This case study comprises of the popular literature example by Kondili et al. (1993) Halim and Srinivasan (2011) adapted this complex production process to include energy and water using aspects, and it is described here. For the production process, two products, Product 1 and Product 2 are to be produced from three raw materials Feed A, Feed B and Feed C. A heater, HR is used to heat Feed A. Two reactors, RR1 and RR2 are available to perform three different chemical reactions, Reaction 1, Reaction 2 and Reaction 3. Finally, a separator exists to purify Impure E. Figure 5.4 shows the STN representation of the problem. Steam is available for heating, cooling water is available for cooling and freshwater is available for washing. The production recipe is as follows:
1. Heating: FeedA is heated from 50°C to 70°C inside HR. Steam is required as a heating medium for this reaction.

2. Reaction 1: A mixture of 50% FeedB and 50% FeedC, on a mass basis, is reacted in either unit RR1 or RR2. The product of this reaction is IntBC. The reaction requires cooling from 100°C to 70°C.

3. Reaction 2: A mixture of 40% Hot A and 60% IntBC is reacted to form Prod1 (40%) and IntAB (60%). This reaction can be performed in either unit RR1 or RR2 and requires heating from 70°C to 100°C.

4. Reaction 3: A mixture of 20% FeedC and 80% IntAB is reacted in either unit RR1 or RR2. The reaction produces ImpureE and requires heating from 100°C to 130°C.

5. Separation: In SR, ImpureE is purified to produce Prod2 (90%) and intermediate IntAB (10%). The separation requires cooling from 130°C to 100°C.

Water is required for washing RR1 and RR2 at the end of any reaction. In this case study, four contaminants, ar, br, cp and dw are present. Table 5.5 shows information regarding the production process. Data pertaining to heating and cooling are given in Table 5.6 while data pertaining to the washing of RR1 and RR2 are given in Table 5.7.

The objective in this case study was to determine the minimum makespan required to produce 200 kg each of Prod1 and Prod2 while also minimising external utility and freshwater consumption.
It is important to note the following from the data in Table 5.5, Table 5.6 and Table 5.7:

1. The processing duration of a batch is dependent on the batch size.
2. The mass load of contaminants is not fixed, but dependent on the batch size.
3. The energy requirements of the operations are a function of the batch size and the given initial and final temperatures. Due to this fact, careful consideration must be given to ensure that whenever heat integration occurs between two operations, the minimum temperature difference for heat transfer, 10°C, is not violated.

### Table 5.5. Data pertaining to production, for case study II

<table>
<thead>
<tr>
<th>Task (i)</th>
<th>Unit (j)</th>
<th>Max batch size (kg)</th>
<th>α_i (h)</th>
<th>β_i (h)</th>
<th>Washing time (h)</th>
<th>Material state</th>
<th>Initial inventory (kg)</th>
<th>Max storage (kg)</th>
<th>Revenue/cost ($/kg or $/MJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heating (H)</td>
<td>HR</td>
<td>100</td>
<td>0.667</td>
<td>0.007</td>
<td>0</td>
<td>FeedA (s1)</td>
<td>1000</td>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td>Reaction 1 (R1)</td>
<td>RR1</td>
<td>50</td>
<td>1.084</td>
<td>0.027</td>
<td>0.25</td>
<td>FeedB (s2)</td>
<td>1000</td>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td>Reaction 1 (R1)</td>
<td>RR2</td>
<td>80</td>
<td>1.034</td>
<td>0.017</td>
<td>0.3</td>
<td>FeedC (s3/s4)</td>
<td>1000</td>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td>Reaction 2 (R2)</td>
<td>RR1</td>
<td>50</td>
<td>1.09</td>
<td>0.027</td>
<td>0.25</td>
<td>HotA (s5)</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Reaction 2 (R2)</td>
<td>RR2</td>
<td>80</td>
<td>1.034</td>
<td>0.017</td>
<td>0.3</td>
<td>IntAB (s8)</td>
<td>0</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>Reaction 3 (R3)</td>
<td>RR1</td>
<td>50</td>
<td>0.417</td>
<td>0.013</td>
<td>0.25</td>
<td>IntBC (s6)</td>
<td>0</td>
<td>150</td>
<td>0</td>
</tr>
<tr>
<td>Reaction 3 (R3)</td>
<td>RR2</td>
<td>80</td>
<td>0.367</td>
<td>0.008</td>
<td>0.3</td>
<td>ImpureE (s9)</td>
<td>0</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>Separation (S)</td>
<td>SR</td>
<td>200</td>
<td>1.334</td>
<td>0.007</td>
<td>0</td>
<td>Prod1 (s7)</td>
<td>0</td>
<td>1000</td>
<td>20</td>
</tr>
<tr>
<td>Separation (S)</td>
<td>SR</td>
<td>200</td>
<td>1.334</td>
<td>0.007</td>
<td>0</td>
<td>Prod2 (s10)</td>
<td>0</td>
<td>1000</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Washwater</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Wastewater</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cooling water</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Steam</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 5.6. Data pertaining to energy requirements, for case study II

<table>
<thead>
<tr>
<th>Task (i)</th>
<th>T_in (°C)</th>
<th>T_out (°C)</th>
<th>Unit (j)</th>
<th>Cp (kJ/kg°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heating (H)</td>
<td>50</td>
<td>70</td>
<td>HR</td>
<td>2.5</td>
</tr>
<tr>
<td>Reaction 1 (R1)</td>
<td>100</td>
<td>70</td>
<td>RR1</td>
<td>3.5</td>
</tr>
<tr>
<td>Reaction 1 (R1)</td>
<td>100</td>
<td>70</td>
<td>RR2</td>
<td>3.5</td>
</tr>
<tr>
<td>Reaction 2 (R2)</td>
<td>70</td>
<td>100</td>
<td>RR1</td>
<td>3.2</td>
</tr>
<tr>
<td>Reaction 2 (R2)</td>
<td>70</td>
<td>100</td>
<td>RR2</td>
<td>3.2</td>
</tr>
<tr>
<td>Reaction 3 (R3)</td>
<td>100</td>
<td>130</td>
<td>RR1</td>
<td>2.6</td>
</tr>
<tr>
<td>Reaction 3 (R3)</td>
<td>100</td>
<td>130</td>
<td>RR2</td>
<td>2.6</td>
</tr>
<tr>
<td>Separation (S)</td>
<td>130</td>
<td>100</td>
<td>SR</td>
<td>2.8</td>
</tr>
<tr>
<td>Cooling water</td>
<td>20</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steam</td>
<td>170</td>
<td>160</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5.7. Data pertaining to water requirements, for case study II

<table>
<thead>
<tr>
<th>Task (i)</th>
<th>Unit (j)</th>
<th>Max inlet concentration (ppm)</th>
<th>Max outlet concentration (ppm)</th>
<th>Contaminant loading (g contaminant/kg batch)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ar</td>
<td>br</td>
<td>cp</td>
</tr>
<tr>
<td>Reaction 1</td>
<td>RR1</td>
<td>300</td>
<td>500</td>
<td>800</td>
</tr>
<tr>
<td>Reaction 2</td>
<td>RR1</td>
<td>700</td>
<td>600</td>
<td>300</td>
</tr>
<tr>
<td>Reaction 3</td>
<td>RR1</td>
<td>500</td>
<td>200</td>
<td>400</td>
</tr>
</tbody>
</table>

The second case study was solved using the proposed formulation, using Constraints (5.1), (5.2), (5.17) to (5.23), (5.26) to (5.35), (5.40) to (5.43), (5.45) to (5.50), (5.52), (5.60) to (5.69) and (5.85) to (5.88). The objective function was the minimisation of makespan, Constraint (5.92). It is important to note that in Constraints (5.40), (5.41), (5.43) and (5.45), variables associated with wastewater storage are not included. This is in order to make a direct comparison with the results obtained by Halim and Srinivasan (2011). The computer used to solve the model had an Intel(R) Core(TM) i7-2670QM, 2.2 GHz processor with 4.0GB RAM. The problem was solved with GAMS using DICOPT for the MINLP with CPLEX as the MIP solver and MINOS as the NLP solver. Figure 5.5 shows the Gantt chart with the resulting production schedule for the case study.
A comparison between the results of this case study obtained by Halim and Srinivasan (2011) and the proposed formulation is described in Table 5.8.

<table>
<thead>
<tr>
<th>Table 5.8. Comparison of results for case study II</th>
<th>Halim &amp; Srinivasan(2011)</th>
<th>This formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Makespan (h)</td>
<td>19.96</td>
<td>19.93</td>
</tr>
<tr>
<td>Steam (MJ)</td>
<td>61.36</td>
<td>44.88</td>
</tr>
<tr>
<td>Cooling water (MJ)</td>
<td>35.39</td>
<td>19.72</td>
</tr>
<tr>
<td>Total freshwater (kg)</td>
<td>275.09</td>
<td>341.2</td>
</tr>
<tr>
<td>Revenue from products ($)</td>
<td>8000</td>
<td>8000</td>
</tr>
<tr>
<td>Cost of raw materials ($)</td>
<td>5604.4</td>
<td>5444.4</td>
</tr>
<tr>
<td>Cost of steam ($)</td>
<td>61.36</td>
<td>44.88</td>
</tr>
<tr>
<td>Cost of cooling water ($)</td>
<td>0.7078</td>
<td>0.3994</td>
</tr>
<tr>
<td>Cost of freshwater ($)</td>
<td>27.509</td>
<td>34.12</td>
</tr>
<tr>
<td>Cost of wastewater ($)</td>
<td>13.75</td>
<td>17.06</td>
</tr>
<tr>
<td>Profit ($)</td>
<td>2292.3</td>
<td>2459.1</td>
</tr>
<tr>
<td>Number of slots</td>
<td>8</td>
<td>N/A</td>
</tr>
<tr>
<td>Number of time points</td>
<td>N/A</td>
<td>17</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>3500</td>
<td>N/A</td>
</tr>
<tr>
<td>CPU time (s)</td>
<td>not reported</td>
<td>24 532</td>
</tr>
<tr>
<td>Number of binary variables</td>
<td>not reported</td>
<td>836</td>
</tr>
<tr>
<td>Major iterations</td>
<td>-</td>
<td>9</td>
</tr>
</tbody>
</table>

As can be observed from Table 5.8, improved objectives were obtained using the proposed formulation with the exception of the amount of freshwater used for washing. This was probably due to the fact that the cost associated with using steam was more significant than the cost associated with using freshwater. Hence, to minimise overall costs, promoting opportunities for heat integration took precedence over promoting opportunities for water reuse. This can be observed from the production schedule in Figure 5.5. With the operations aligned to promote heat integration, no opportunities for water reuse exist at all.

The MINLP model was solved by the aforementioned initialisation procedure. The objective value from the relaxed MINLP model was 123.543 $/h and the objective value from the exact MINLP model was 123.369 $/h. As both the relaxed model and exact models were MINLP, due to the nonlinear objective function, the global optimality of the solution could not be guaranteed.
5.6. Case study III

This case study was obtained from Majozi and Gouws (2009) and was extended to include heat integration opportunities. The multipurpose batch facility investigated, is similar to that discussed in Case Study II. The heating and separation tasks performed in HR and SR respectively are not to be heat integrated with any other units. Heat integration can only occur between RR1 and RR2 depending on the tasks they perform. Similarly, water is required for washing RR1 and RR2 at the end of any reaction. In this case study, three contaminants, C1, C2 and C3 are present. Table 5.9 shows information regarding the production process, while data pertaining to the washing of RR1 and RR2, obtained from Majozi and Gouws (2009) are given in Table 5.10. The heat integration data is provided in Table 5.11.

Table 5.9. Production data for case study III

<table>
<thead>
<tr>
<th>Task (i)</th>
<th>Unit (j)</th>
<th>Max batch size (kg)</th>
<th>Mean processing time (h)</th>
<th>Washing time (h)</th>
<th>Material state</th>
<th>Initial inventory (kg)</th>
<th>Max storage (kg)</th>
<th>Revenue/cost (cost units/kg or cost units/kJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heating (H)</td>
<td>HR</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>FeedA (s1)</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>0</td>
</tr>
<tr>
<td>Reaction 1 (R1)</td>
<td>RR1</td>
<td>50</td>
<td>2</td>
<td>0.25</td>
<td>FeedB (s2)</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>80</td>
<td>2</td>
<td>0.3</td>
<td>FeedC (s3/s4)</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>0</td>
</tr>
<tr>
<td>Reaction 2 (R2)</td>
<td>RR1</td>
<td>50</td>
<td>2</td>
<td>0.5</td>
<td>HotA (s5)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>80</td>
<td>2</td>
<td>0.25</td>
<td>IntAB (s8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Reaction 3 (R3)</td>
<td>RR1</td>
<td>50</td>
<td>1</td>
<td>0.25</td>
<td>IntBC (s6)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>80</td>
<td>1</td>
<td>0.25</td>
<td>ImpureE (s9)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Separation (S)</td>
<td>SR</td>
<td>200</td>
<td>1 for Prod2</td>
<td>0</td>
<td>Prod1 (s7)</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 for IntAB</td>
<td></td>
<td>Prod2 (s10)</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Washwater</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Wastewater</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cooling water</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Steam</td>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5.10. Wastewater minimisation data for case study III

<table>
<thead>
<tr>
<th>Reaction Type</th>
<th>Maximum contaminant concentration (g contaminant/kg water)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C1</td>
</tr>
<tr>
<td>Reaction 1 (RR1) Max. inlet</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Max. outlet</td>
</tr>
<tr>
<td>Reaction 2 (RR1) Max. inlet</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Max. outlet</td>
</tr>
<tr>
<td>Reaction 3 (RR1) Max. inlet</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>Max. outlet</td>
</tr>
<tr>
<td>Reaction 1 (RR2) Max. inlet</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>Max. outlet</td>
</tr>
<tr>
<td>Reaction 2 (RR2) Max. inlet</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>Max. outlet</td>
</tr>
<tr>
<td>Reaction 3 (RR2) Max. inlet</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>Max. outlet</td>
</tr>
</tbody>
</table>

Table 5.11. Heat integration data for case study III

<table>
<thead>
<tr>
<th>Reaction Type</th>
<th>Heating/cooling requirement (kWh)</th>
<th>Operating Temperature (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX1 exothermic</td>
<td>60 (cooling)</td>
<td>100</td>
</tr>
<tr>
<td>RX2 endothermic</td>
<td>80 (heating)</td>
<td>60</td>
</tr>
<tr>
<td>RX3 exothermic</td>
<td>70 (cooling)</td>
<td>140</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heat storage parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cp_{fluid} (kJ/kg°C)</td>
<td>4.2</td>
</tr>
<tr>
<td>ΔT^{min} (°C)</td>
<td>10</td>
</tr>
<tr>
<td>T^L (°C)</td>
<td>20</td>
</tr>
<tr>
<td>T^U (°C)</td>
<td>180</td>
</tr>
<tr>
<td>W^L (ton)</td>
<td>1</td>
</tr>
<tr>
<td>W^U (ton)</td>
<td>3</td>
</tr>
</tbody>
</table>
In this case study, storage facilities for heat storage and water reuse are available. Different scenarios of the case study were solved to demonstrate the capabilities of the model. These scenarios are as follows:

- **Scenario 1**: During production, only freshwater is available for washing. Heating and cooling are provided exclusively by steam and cooling water.
- **Scenario 2**: In addition to freshwater for washing, steam and cooling water for heating and cooling respectively, opportunities for direct water reuse and direct heat integration are explored.
- **Scenario 3**: The effect of the inclusion of storage facilities for heat and wastewater (indirect water reuse and indirect heat integration) to Scenario 2 is explored.

Constraints (5.1) to (5.16), (5.24) to (5.25), (5.34) to (5.39), (5.40) to (5.43), (5.44) and (5.47) to (5.91) were used to solve this case study. The bilinear terms present in the model were linearised and the solution procedure as described above was used.

The capacity of the storage vessel for water was 200 kg. The objective of this case study was to maximise profit while minimising wastewater production and energy consumption, within a time horizon of 12 h. The storage capacity and the initial storage temperature of the storage vessel were variables to be optimised (Stamp and Majozi, 2011). Heat losses were not considered.

The computer used to solve the model had an Intel(R) Core(TM) i7-2670QM, 2.2 GHz processor with 4.0GB RAM. The problem was solved with GAMS using DICOPT for the MINLP with CPLEX as the MIP solver and CONOPT as the NLP solver. A comparison between the three scenarios is provided in Table 5.12. The results of Scenario 1, Scenario 2 and Scenario 3 are contained in column 2, 3 and 4 respectively, of Table 5.12.
Table 5.12. Comparison between different scenarios for case study III

<table>
<thead>
<tr>
<th></th>
<th>Freshwater and utilities</th>
<th>Direct water reuse and direct heat integration</th>
<th>Direct/indirect water reuse and direct/indirect heat integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profit (c.u.)</td>
<td>18 537</td>
<td>19 836</td>
<td>22 235</td>
</tr>
<tr>
<td>Amount of Prod1 (kg)</td>
<td>96</td>
<td>116</td>
<td>116</td>
</tr>
<tr>
<td>Amount of Prod2 (kg)</td>
<td>162</td>
<td>156.4</td>
<td>188</td>
</tr>
<tr>
<td>Cooling water (kWh)</td>
<td>390</td>
<td>250</td>
<td>190</td>
</tr>
<tr>
<td>Steam (kWh)</td>
<td>240</td>
<td>180</td>
<td>10</td>
</tr>
<tr>
<td>Freshwater (kg)</td>
<td>816</td>
<td>1 020</td>
<td>896</td>
</tr>
<tr>
<td>Time points</td>
<td>11</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>CPU time (s)</td>
<td>3.1</td>
<td>14.8</td>
<td>42 275</td>
</tr>
<tr>
<td>Binary variables</td>
<td>128</td>
<td>508</td>
<td>954</td>
</tr>
<tr>
<td>Initial storage temperature (ºC)</td>
<td></td>
<td></td>
<td>82.9</td>
</tr>
<tr>
<td>Heat storage capacity (ton)</td>
<td></td>
<td></td>
<td>2.024</td>
</tr>
<tr>
<td>Major iterations</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

From Table 5.12 it can be observed that profit increases from Scenario 1 to Scenario 3, with a decrease in steam and cooling water requirements. However, while cooling water and steam decreased in Scenario 2 compared to Scenario 1, the amount of freshwater increased. The total amount of product in Scenario 1 is 258 kg while the total amount of product in Scenario 2 is 272.4 kg. This is as a result of additional unit operations being performed in Scenario 2 than are performed in Scenario 1. These additional unit operations contribute to the increased amount of washing water required. Furthermore, no opportunities for direct water reuse were realised due to the cost of steam relative to the cost of freshwater. The unit operations are aligned in such a way as to promote as much heat integration as possible. In so doing, opportunities for direct water reuse are lost. In Scenario 3, storage for water is available and hence a decrease is observed in the amount of freshwater used. The resulting process schedule for the results of Scenario 2 is illustrated in Figure 5.6 while the corresponding schedule for Scenario 3 is illustrated in Figure 5.7.
Figure 5.6. Resulting process schedule when only direct heat integration and direct water reuse are possible

The clear blocks represent the production operation in the unit while the dark blocks represent the washing operations which take place after the reactions are completed. The numbers above the clear blocks represent the amount of material processed in the unit during production and the numbers below the washing operations represent freshwater. Water transfer to and from storage has been clearly labelled. The up-down arrows represent direct heat integration while the bent arrows represent indirect heat integration to or from the heat storage unit. The results of scenario 1 are globally optimal with the objective function of the linearised MILP and exact MINLP being 18 537 c.u. Similarly, the results of scenario 2 are globally optimal, with the objective function of the linearised MILP and exact MINLP being 19 836 c.u. In scenario 3, the objective function of the linearised MILP was 25 270 c.u, while the objective function of the exact MINLP was 22 235 c.u. Hence, the results for scenario 3 are locally optimal.
5.7. Conclusions

A simultaneous method for the optimisation of energy and water embedded within a scheduling framework has been developed. Furthermore, opportunities for direct and indirect heat integration as well as direct and indirect water reuse have been explored. The mathematical formulation led to an MINLP problem for which an initialisation procedure was employed. The applicability of the method has been demonstrated with three case studies. The developed formulation has proved to effectively solve a complex makespan minimisation problem in which duration is a function of batch size and which included multiple contaminants. The advantage of the developed model is that the variable nature of time ensures that the schedule is flexible and results that could have been overlooked when time is fixed are available during the optimisation search, leading to better overall objectives. In future, the problem can be extended to include other water using operations other than equipment washing. Furthermore, more efficient scheduling models can be adopted to investigate the effect on objective values and computational time.
Nomenclature

Sets

\[
\begin{align*}
C &= \{ c \mid c = \text{contaminant} \} \\
J &= \{ j \mid j = \text{processing unit} \} \\
J_c &= \{ j_c \mid j_c = \text{processing unit which may conduct tasks requiring heating} \} \subseteq J \\
J_h &= \{ j_h \mid j_h = \text{processing unit which may conduct tasks requiring cooling} \} \subseteq J \\
J_w &= \{ j_w \mid j_w = \text{processing unit which requires washing after conducting a task} \} \subseteq J \\
P &= \{ p \mid p = \text{time point} \} \\
S &= \{ s \mid s = \text{any state} \} \\
S_{in} &= \{ s_{in} \mid s_{in} = \text{input state into any unit} \} \\
S_{in,j} &= \{ s_{in,j} \mid s_{in,j} = \text{input state to a processing unit} \} \subseteq S \\
U &= \{ u \mid u = \text{heat storage unit} \}
\end{align*}
\]

Continuous variables

\[
\begin{align*}
B(s_{in,j}, p) &= \text{batch size, either fixed or variable} \\
c_{in}(s_{in,j}, c, p) &= \text{inlet concentration of contaminant } c, \text{ to unit } j \in J_w \text{ at time point } p \\
c_{out}(s_{in,j}, c, p) &= \text{outlet concentration of contaminant } c, \text{ from unit } j \in J_w \text{ at time point } p \\
CL(s_{in,j}, p) &= \text{cooling load for hot state} \\
CL'(s_{in,j}, p) &= \text{cooling load per time, for hot state} \\
c_{in}(c, p) &= \text{inlet concentration of contaminant } c, \text{ to storage at time point } p \\
c_{out}(c, p) &= \text{outlet concentration of contaminant } c, \text{ from storage at time point } p \\
cw(s_{in,j}, p) &= \text{external cooling required by unit } j_h \text{ conducting the task corresponding} \\
&\quad \text{to state } s_{in,j_h} \text{ at time point } p \\
d(s, p) &= \text{amount of state delivered to customers at time point } p \\
dur(s_{in,j}, p) &= \text{duration of task, dependent on batch size}
\end{align*}
\]
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\( F_e (s_{in,j}, p) \)  mass of effluent water from unit \( j \in J_w \) at time point \( p \)

\( F_f (s_{in,j}, p) \)  mass of freshwater into unit \( j \in J_w \) at time point \( p \)

\( F_{in} (s_{in,j}, p) \)  mass of water into unit \( j \in J_w \) for washing at time point \( p \)

\( F_{out} (s_{in,j}, p) \)  mass of water exiting unit \( j \in J_w \) at time point \( p \) after washing

\( F_r (s_{in,j}, s_{in,j}', p) \)  mass of water recycled/reused from washing unit \( j \) after processing \( s_{in,j} \) to unit \( j' \) after processing \( s_{in,j} \) at time point \( p \)

\( F_{s_{in}} (s_{in,j}, p) \)  mass of water transferred from unit \( j \in J_w \) to storage at time point \( p \)

\( F_{s_{out}} (s_{in,j}, p) \)  mass of water transferred from storage to unit \( j \in J_w \) at time point \( p \)

\( H \)  time horizon of interest, optimisation variable for makespan minimisation problem

\( HL (s_{in,j}, p) \)  heating load for cold state

\( \dot{H}L (s_{in,j}, p) \)  heating load per time, for cold state

\( M_B (s_{in,j}, c, p) \)  mass load of contaminant \( c \) in unit \( j \in J_w \) at time point \( p \) after processing \( s_{in,j} \) that is added to the water stream

\( m_p (s_{in,j}, p) \)  amount of state produced after processing task \( s_{in,j} \) at time point \( p \)

\( m_p (s_{in,j}, p) \)  amount of state consumed to process task \( s_{in,j} \) at time point \( p \)

\( Q (s_{in,j}, u, p) \)  heat exchanged with heat storage unit \( u \) at time point \( p \)

\( q (s_{in,j}, s_{in,j}, p) \)  amount of heat exchanged during direct heat integration

\( q_s (s, p) \)  amount of state \( s \) stored at time point \( p \)

\( qw_s (p) \)  amount of water stored in storage at time point \( p \)

\( s_d (s_{in,j}, p) \)  external heating required by unit \( j \) conducting the task corresponding to state \( s_{in,j} \) at time point \( p \)

\( T_i (u, p) \)  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 \( p \)

\( t_i (s_{in,j}, u, p) \)  time at which heat storage unit commences activity

\( t_f (s_{in,j}, u, p) \)  time at which heat storage unit ends activity

\( t_{out} (s_{in,j}, p) \)  time at which a state is produced from unit \( j \) at time point \( p \)
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\[ t_u(s_{in,j}, p) \] time at which a state is used in unit \( j \)

\[ ts_{in}(s_{in,j}, p) \] time at which water is transferred from unit \( j \in J_w \) to storage at time point \( p \)

\[ ts_{out}(s_{in,j}, p) \] time at which water is transferred from storage to unit \( j \in J_w \) at time point \( p \)

\[ tw_{in}(s_{in,j}, p) \] time at which water enters unit \( j \in J_w \) at time point \( p \)

\[ tw_{out}(s_{in,j}, p) \] time at which water exits unit \( j \in J_w \) at time point \( p \)

\[ tw_r(s_{in,j}, s_{in,j'}, p) \] time at which water is recycled from unit \( j \) to unit \( j' \) \((j, j' \in J_w)\) at time point \( p \)

\[ W(u) \] capacity of heat storage unit \( u \)

\[ \Gamma(s_{in,j}, u, p) \] Glover Transformation variable

\[ \Psi(s_{in,j}, u, p) \] Reformulation-Linearisation variable

Binary variables

\[ x(s_{in,j}, s_{in,j'}, p) \] binary variable associated with heat integration between unit \( j_c \) conducting the task corresponding to state \( s_{in,j_c} \) and unit \( j_h \) conducting the task corresponding to state \( s_{in,j_h} \) at time point \( p \)

\[ y(s_{in,j}, p) \] binary variable associated with usage of state \( s \) in unit \( j \) for production at time point \( p \)

\[ y_{in}(s_{in,j}, p) \] binary variable showing transfer of water from unit \( j \in J_u \) to storage at time point \( p \)

\[ y_{out}(s_{in,j}, p) \] binary variable showing transfer of water from storage to unit \( j \in J_w \) at time point \( p \)

\[ y_{w}(s_{in,j}, p) \] binary variable showing usage of water in unit \( j \in J_w \) at time point \( p \)

\[ y_{w_r}(s_{in,j}, s_{in,j'}, p) \] binary variable showing reuse/ recycle of water from unit \( j \) to unit \( j' \) \((j, j' \in J_w)\) at time point \( p \)

© University of Pretoria
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\[ z(s_{in,j}, u, p) \] binary variable associated with heat integration between unit \( j \) conducting the task corresponding to state \( s_{in,j} \) with heat storage unit \( u \) at time point \( p \).

Parameters

\( C_{in}^U(s_{in,j}, c) \) maximum inlet concentration of contaminant \( c \) in unit \( j \in J_w \)

\( C_{out}^U(s_{in,j}, c) \) maximum outlet concentration of contaminant \( c \) from unit \( j \in J_w \)

\( CE \) cost of effluent water treatment

\( CF \) cost of freshwater

\( CL(s_{in,j}) \) fixed cooling load for hot state

\( Cost_cw \) cost of cooling water

\( Cost_st \) cost of steam

\( Cp_{fluid} \) specific heat capacity of heat storage fluid

\( Cp_{state}(s_{in,j}) \) specific heat capacity of state

\( CP(s) \) selling price of product \( s \), \( s = \text{product} \)

\( CS^w_{out}(c) \) initial concentration of contaminant in storage

\( F^U(s_{in,j}) \) maximum inlet water mass of unit \( j \in J_w \)

\( H \) time horizon of interest, a parameter for profit maximisation problem

\( HL(s_{in,j}) \) fixed heating load for cold state

\( M(s_{in,j}, c) \) mass load of contaminant \( c \) in unit \( j \in J_w \) after processing \( s_{in,j} \) that is added to the water stream

\( MM \) any large number

\( \Delta M(s_{in,j}, c) \) contaminant loading

\( Q^{\text{max}}(s_{in,j}) \) maximum possible heating load or cooling load for a cold state or hot state, respectively

\( Q^w(s) \) initial amount of state \( s \) in storage

\( Q^U(s) \) maximum capacity of storage for state \( s \)

\( Q_{w}^o \) initial amount of water in storage
Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework

\[ Q_{W_j}^U \]  
maximum capacity of storage for water

\[ T(s_{in,j}) \]  
operating temperature for processing unit \( j \) conducting the task corresponding to state \( s_{in,j} \), for constant temperature processes

\[ T^L \]  
lower bound for heat storage temperature

\[ T^U \]  
upper bound for heat storage temperature

\[ T_{in}(s_{in,j}) \]  
inlet temperature for state \( s_{in,j} \)

\[ T_{out}(s_{in,j}) \]  
outlet temperature for state \( s_{in,j} \)

\[ \Delta T_{min} \]  
minimum allowable thermal driving force

\[ V_j^{min} \]  
iminimum capacity of a unit

\[ V_j^{max} \]  
imaximum capacity of a unit

\[ W_j^L \]  
lower bound for heat storage capacity

\[ W_j^U \]  
upper bound for heat storage capacity

\[ \alpha \]  
constant coefficient of processing time

\[ \beta \]  
variable coefficient of processing time

\[ \tau(s_{in,j}) \]  
duration of the task corresponding to state \( s_{in,j} \) conducted in unit \( j \)

\[ \tau_w(s_{in,j}) \]  
duration of washing for unit \( j \in J_w \)

References


Chapter 5. A Unified Approach for the Optimisation of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework


CHAPTER 6
Conclusions and Recommendations

6.1. Conclusions

This thesis presents three mathematical formulations for multipurpose batch plants based on a flexible production schedule, unit specific, continuous time and state sequence network representations. The first addresses sequence dependent changeover costs. The second investigates wastewater minimisation as a consequence of the appropriate sequencing of tasks in units in determining the occurrence and extent of washing. The third considers the simultaneous optimisation of energy and water to reduce costs and improve profit. These investigations are in response to gaps observed in the literature, problems that have not been previously dealt with or can be improved upon.

The prerequisite for a model to address sequence dependent changeover costs is the ability to correctly establish the immediate successors of all tasks in each unit. This is not a trivial problem for multipurpose batch plants based on unit specific, continuous time and state network representations due to the fact that although tasks occur consecutively in the same unit, they do not necessarily take place at consecutive time points. That is, the difference between the time points of two consecutive tasks on a unit may be greater than 1. In addition, recent advancements made regarding the development of robust scheduling formulations for short-term production have largely ignored sequence dependent changeover costs.

The presented MILP formulation for short-term production scheduling including sequence dependent changeover costs was developed to accomplish this. The formulation addressed additional constraints in the form of fixed utility availability, the absence of storage for intermediate material (NIS) and the possibility of intermediate storage in any idle unit. When applied to a case study, the formulation produced a smaller model while achieving the same objective when compared to an existing formulation. This resulted in a shorter computational time.
Chapter 6. Conclusions and Recommendations

Following this accomplishment, the explicit consideration of resources required for changeover, water in this case, was investigated. Water used for washing is an additional cost and generated wastewater is another consequence. The proposed mathematical formulation minimised the occurrence of equipment washing and the amount of water required, based on the sequence of tasks taking place in a unit, while still meeting the production objectives. Three scenarios of the problem were investigated. The first determined the water requirement if water use was independent of the sequence of tasks. The second considered the effect of proper sequencing on the washing requirement. The third scenario proposed the possibility of allowing a degree of contamination in equipment units for closely related tasks. The formulation was applied to two case studies and a 48% reduction in water usage was observed in Case study II involving a multipurpose batch plant.

The formulation for minimising wastewater as a result of proper task sequencing was extended to include process integration in the form of direct water reuse. When applied to a case study, a 25% reduction in wastewater generated was observed compared to sequence independent washing with direct water reuse. The different formulations were MILP models.

Finally, a mathematical formulation was presented to demonstrate the capability of a unified model that incorporated the optimisation of energy, water and batch production scheduling in one framework. It proved superior to a methodology that adopted a sequential path to the solution in terms of the objective values obtained. A reason for this was that because the proposed formulation is truly simultaneous, results that could have been omitted in a sequential approach are available in the search space during optimisation.

Furthermore, the robustness of the model was demonstrated by the introduction of indirect water reuse and indirect heat integration opportunities. As expected, these opportunities led to an increase in profit with a 20% improvement observed with heat storage and water storage. However, the increased complexity of the model from the additional constraints led to an MINLP model requiring a reformulation-linearisation procedure to solve.
6.2. **Recommendations for future work**

Scheduling formulations are most useful when constraints can describe real life plant conditions as closely as possible. As the case studies considered in this thesis have been academic, it is recommended that the formulations be applied to real industrial problems. A consequence of this is that certain assumptions will be discarded. This suggests that multiple contaminants should be considered during wastewater minimisation, mixed batch and continuous processes be included and uncertainty during batch production be investigated.

It is also recommended that heat integration and wastewater minimisation be investigated in the context of costs required for additional heat exchangers to facilitate heat recovery and pipes and pumps to facilitate water reuse. Furthermore, other interactions between water and energy apart from washing water required after a task has ended and the energy associated with heating or cooling the same task should be investigated.

Finally, the formulation for wastewater minimisation that takes advantage of the sequence of tasks in a unit to determine the extent of water-using operations is a great foundation for the incorporation of further process integration techniques such as water storage and regeneration. For example, how will appropriate task sequencing determine the capacity of water storage required?