WASTEWATER MINIMIZATION IN MULTIPURPOSE BATCH PROCESSES USING MATHEMATICAL MODELLING

Muimeleli Mutangwa (535277)

A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in fulfillment of the requirements for the degree of Master of Science in Engineering

Supervisor: Professor Thokozani Majozi

May 2018

Declaration

I declare that this dissertation is my own unaided work. It is being submitted for the Degree of Master of Science in Chemical Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other university.

....

(Signature of Candidate)

.... day of .... year........................

Acknowledgements

My deepest gratitude goes to my role model and supervisor, Prof T. Majozi. You made this journey an experience that I shall cherish for the rest of my life. Besides your guidance and patience; you gave me an opportunity to travel abroad for the first time to present at a conference in China where this work received an award.

I would like to thank my colleagues at the Sustainable Process Engineering department (Wits University) for maintaining a supportive environment that encourages research.

I acknowledge the funders of this work: the National Research Foundation, through the DST Chair in Sustainable Process Engineering, and the Water Research Commission.

I am also grateful to my family and friends for their unconditional love and for believing in me.

Dear God, Thank you!

Synopsis

The increase in the degradation of water sources and stringent environmental regulations have greatly motivated industries to explore means of utilizing water efficiently. Batch processes are known to generate highly contaminated wastewater that is toxic to the environment. A holistic approach to design which emphasizes the unity of the process, process integration (PI), can be used to reduce both the wastewater generated and the level of contamination while maintaining the profitability of the chemical plant. Process integration techniques for wastewater minimization in batch processes include water reuse, recycle and regeneration.

Most mathematical formulations for wastewater minimization in multipurpose batch processes presented in literature determine the amount of water required for washing operations by only looking at the task that has just occurred in a unit. However, the nature of the succeeding task can influence the amount of water required for the washing operation between consecutive tasks in a processing unit. In paint manufacturing, for example, more water will be required for the washing operation if the production of white paint follows the production of black paint and less water will be required if the black paint follows the white paint. The amount of wastewater generated in batch processes can, therefore, be reduced by simply synthesizing a sequence of tasks that will generate the least amount of wastewater. Presented in this work are wastewater minimization formulations for multipurpose batch processes which explore sequence dependent changeover opportunities for water minimization simultaneously with direct and indirect water reuse and recycle opportunities.

The presence of continuous and integer variables, as well as bilinear terms, rendered the model a Mixed Integer Nonlinear Program (MINLP). The developed MINLP model was validated using two single contaminant illustrative examples and a multiple contaminant example. A global optimization solver, Branch and Reduce Optimization Navigator (BARON), was used to solve the optimization problems on a

General Algebraic Modeling System (GAMS) platform. Exploring multiple water saving opportunities simultaneously has proven to be computationally intensive but can result in significant water savings. For instance, two different scenarios saved 65% and 61% in freshwater use respectively.

Table of contents

List of figures

List of tables

List of abbreviations

BARON	Branch and Reduce Optimization Navigator
CIS	Common Intermediate Storage
c.u	Cost Unit
FIS	Finite Intermediate Storage
FW	Finite Wait
g	Grams
GAMS	General Algebraic Modelling System
GBD	General Benders Decomposition
H	Time Horizon
hr	Hours
Kg	Kilograms
L	Liters
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Nonlinear Programming
MIS	Mixed Intermediate Storage
NIS	No Intermediate Storage
NLP	Nonlinear Programming
OA	Outer Approximation

PIS	Process Intermediate Storage
R	Reactor
RTN	Resource Task Network
RU	Regeneration Unit
Rxn	Reaction
s	Seconds
SSN	State Sequence Network
ST	Storage Tank
STN	State Task Network
TAM	Time Average Models
U	Unit
UIS	Unlimited Intermediate Storage
UW	Unlimited Wait
ZW	Zero Wait
WEF	World Economic Forum
NWRS	National Water Research Strategy
NDP	National Development Plan

Chapter 1

INTRODUCTION

1.1. Background

The global water crisis was ranked as the highest risk in 2016 by the World Economic Forum (WEF) and it is one of the biggest threats facing the planet over the next decade (WEF, 2016). The United Nations 2030 Agenda for Sustainable Development, which was drafted to address urgent global challenges, includes ensuring availability and sustainable management of water (United Nations, 2016). The second edition of the National Water Research Strategy (NWRS) which responds to the vision of South Africa for 2030, as articulated by the National Development Plan (NDP), recognized that the socio-economic growth will be restricted if water security and associated water management issues are not resolved in time (NWRS, 2013). According to the WEF (2016), South Africa is the 30th driest country in the world and has less water per person than countries widely considered to be much drier, such as Namibia and Botswana. Industrial processes consume up to 17% of the available water in South Africa, and as a result, significant responsibility for conservation lies with process industries (Council for Scientific and Industrial Research, 2010).

Besides water consumption, industrial processes also degrade water sources. Many industries dispose their wastewater directly into rivers. Water sources that appeal to life on earth are safe, contain no harmful chemical substances and are stable in terms of corrosion. According to Rand Water (2017), freshwater in South Africa is decreasing in quality because of the increase in pollution caused by mining, manufacturing industry, agriculture, etc. Industries produce wastewater that affects the pH of the water, amount of nutrients (causing eutrophication), temperature (impacting temperature-sensitive organisms), and increases murkiness (blocking fish grills, hindering photosynthesis and causing diseases). Wastewater with chemicals that are not found naturally in the environment, or are found in very small amounts, end up poisoning plants, animals and people.

Batch processes have become a popular mode of manufacturing due to their adaptability to volatile conditions that have characterized recent times. Market demands have changed significantly and high value-added products are required in small volumes. Pharmaceutical products, detergents, paints, deodorants, etc., are examples of products that are manufactured using batch plants. Batch processes follow a series of discrete tasks and are getting attention due to their ability to allow for the production of a variety of products that follows different production recipes in one production facility. The nature of batch manufacturing allows for batches of different tasks to share processing units. Washing operations are essential in batch processes since the integrity of each batch needs to be preserved. These washing operations are the major source of wastewater in most batch processes. Although most batch plants generate fewer quantities of wastewater compared to their continuous counterparts, effluents from batch facilities are mostly toxic (Majozi, 2010). The need for investigating water saving measures for batch manufacturing industries was triggered by a combination of the recent public awareness of the impact of industrial pollution on water sources, stringent environmental regulations, and the scarcity of freshwater as a natural resource.

Most production facilities make use of the end-of-pipe treatment as a means of handling wastewater. An end-of-pipe treatment is when all the generated wastewater is sent to a treatment facility. Depending on the nature of the contaminant in the wastewater, treatment methods are divided into physical, chemical and biological. Water is treated such that it meets the required contaminant levels before it is discharged to the environment. Significant financial investment is required for this approach and the cost is highly influenced by the amount of wastewater to be treated. It is therefore logical to explore wastewater minimizing opportunities before sending the wastewater for end-of-pipe treatment. Process integration is an approach for process optimization through emphasizing the unity of the process, environmental issues and process objectives such as profitability (El-Halwagi, 1998). This approach looks at the whole manufacturing process as an integrated system of interconnected processing units as well as utilities and waste streams. Process integration techniques for wastewater minimization in multipurpose batch processes presented in the literature include direct reuse or recycle, indirect reuse or recycle, and regeneration reuse or recycle (Gouws et al., 2010). In this work, multiple water saving opportunities will be explored simultaneously.

1.2. Motivation

Most mathematical models, in literature, for wastewater minimization in batch processes determine the amount of water required for washing operations by only looking at the task that has just taken place in a unit. However, the amount of water required for washing operations can depend on the sequence of tasks in a unit. The amount of water required for washing operations should, therefore, be determined by looking at both the task that takes place in a unit and its successor. As shown in Figure 1.1, the amount of water required for washing the unit when task *B* follows task *A* is not the same as the amount that is required when task *A* follows task *B*. A practical example will be a multipurpose unit that processes black paint and white paint. Due to the sensitivity of the white paint, more water will be required for the washing operation if the white paint follows the black paint and less water will be required if the black paint follows the white paint. Sequence dependent changeover opportunity for water minimization can, therefore, be explored by simply synthesizing the sequence of tasks that optimizes the trade-off between the production and the amount of wastewater generated.

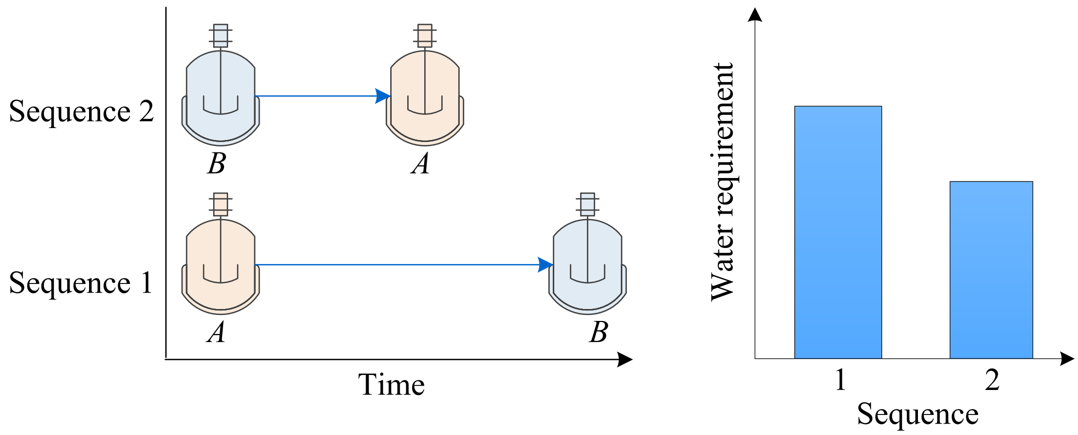


Figure 1.1 Sequence dependent washing water requirement

Adekola and Majozi (2017) developed a mathematical model for simultaneous optimization 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. Since a sequence dependent parameter is required, the formulation presented by Adekola and Majozi (2017) explores sequence dependent opportunities for water minimization in multipurpose batch processes by fixing sequence dependent changeover times. To successfully incorporate sequence dependent constraints, their formulation is able to successfully determine a task that immediately follows the task that has just occurred in a unit. However, the work of Adekola and Majozi (2017) did not explore water reuse and recycle opportunities.

This work aims to develop a mathematical model for the simultaneous optimization of batch scheduling and wastewater minimization where sequence dependent changeover opportunities are explored simultaneously with direct and indirect water reuse and recycle in the presence of a central reusable water tank.

1.3. Objectives

The objectives of the study are as follows:

- To develop mathematical models that explore sequence dependent water saving opportunities.
- To develop mathematical models that explore sequence dependent water saving opportunities simultaneously with direct and indirect water reuse and recycle in the presence of a central storage water tank.
- To validate the developed mathematical formulations using illustrative examples.

1.4. Problem statement

The problem addressed in this study can be stated as follows

Given:

(i) Scheduling data, i.e. product recipe, capacities for different units and suitability, storage capacities, task processing times, time horizon, value of raw materials, products and utilities;

(ii) Water usage data, i.e. concentration of processed material that remains in the unit, inlet and outlet contaminant concentration limits, flowrates, and capacity of central water storage;

(iii) Sequence dependent changeover parameters.

It is required to determine the optimum sequence of tasks in each unit that generates the least amount of wastewater within the time horizon of interest, the minimum amount of freshwater use, the maximum product throughput, and water reuse network.

1.5. Dissertation structure

Chapter 1 introduces the research study by presenting the background followed by the motivation of the study. In this chapter, the problem statement and the scope of the study are stated. The background upon which the research was conducted and the models built, is provided in Chapter 2 through a review of relevant literature. Chapter 3 is model development where the relevant models are presented in detail. Chapter 4 shows results obtained when the developed formulations were applied to two single contaminant illustrative examples and a multiple contaminant example. The limitations of the model are discussed in Chapter 5 together with the recommendations that may influence future research. Conclusions made are presented in Chapter 6.

1.6. References

Adekola, O., Majozi, T., 2017. Wastewater minimization in batch plants with sequence dependent changeover. Comput. Chem. Eng. 97, 85–103. doi:10.1016/j.compchemeng.2016.11.016

El-Halwagi, M.M., 1998. Pollution prevention through process integration. Clean Prod. Process. 1, 5–19.

Gouws, J., Majozi, T., Foo, D.C.Y., Chen, C.L., Lee, J.Y., 2010. Water minimization techniques for batch processes. Ind. Eng. Chem. Res. 49, 8877–8893. doi:10.1021/ie100130a

Majozi, T., 2010. Batch Chemical Process Integration. doi:10.1007/978-90-481-2588-3

NWRS, 2013. Water for an Equitable and Sustainable Future. Deparment Water Aff.

Rand Water, 2017. Water pollution and your health [WWW Document]. URL http://www.randwater.co.za/corporateresponsibility/wwe/pages/waterpollution.a spx

United Nations, 2016. The Sustainable Development Goals Report.

WEF, 2016. The Global Risks Report 2016 11th Edition. World Econ. Forum.

Chapter 2

LITERATURE REVIEW

2.1. Introduction

This chapter presents a review of the literature that forms the basis of the conducted research. A brief outline on process integration is given, followed by a review of batch processes and scheduling techniques since the two are inherently linked to each other. This chapter also assesses previous studies conducted on wastewater minimization in batch processes and ways of handling sequence dependent changeovers. A background work on mathematical optimization and linearization of different nonlinear terms is presented to usher understanding of how complex mathematical problems can be solved.

2.2. What is process integration?

Process integration is a holistic approach that emphasizes the unity of a process with the aim of making efficient use of process equipment, energy, water and other utilities in order to optimize value (El-Halwagi, 2012). This approach for efficient management of resources is useful in industrial processes where raw materials, utilities, products, and effluents are often linked in one way or the other. This observation cannot be explored by analytical approaches that optimize units individually, and this makes process integration approaches superior.

Process integration techniques can be explored during the design stage of a process plant in order to develop a more sustainable design with efficient energy and water systems (Huang et al., 1999). The performance of an already existing processing plant can also be improved through process integration techniques. When focusing on wastewater minimization in production industries, process integration techniques which can be considered include water reuse and recycle.

Process integration techniques are implemented in conjunction with optimization techniques such as graphical techniques, heuristic methods, and mathematical optimization. Graphical techniques are two-dimensional and therefore can only be used for single contaminant problems and cannot handle time as a variable. In heuristic methods, some of the parameters defining a mathematical problem are random. Heuristic methods are considered as a shortcut and do not guarantee optimality. Even though mathematical programming can sometimes yield computationally intensive models; they can, however, handle more complex problems including those with multiple contaminants and where time is treated as a variable.

According to Edgar and Himmelblau (1989), mathematical optimization problems are formulated such that they consist of two essential parts i.e. the process model and at least one objective function. The following demonstrates a structure of a mathematical optimization problem:

Objective: Minimize $f(x)$

Subject to: $g(x) \leq 0$

$H(x) = 0$

The objective function is an expression to be minimized or maximized subject to various variables and constraints described in the process model. The process model describes the physical laws and the interrelationships of the key variables that apply to a specific problem. Mathematical programming is used as a tool to achieve the desired objective by exploring process integration techniques.

2.3. Introduction to batch processes

Batch processes have been receiving attention in recent decades because of the increased market demands of high value-added products and specialty chemicals. Well established design techniques have been developed for continuous processes and most batch processes have been poorly designed (Smith, 2014). Techniques for continuous processes cannot be directly adopted for batch processes due to the additional time dimension that makes batch processes more complex.

A manufacturing process where a recipe, i.e. a predefined sequence from raw materials to desired products, follows a series of discrete tasks is called a batch process (Majozi, 2010). A batch reactor is distinct from a continuous reactor because it is characterized by the discreteness of tasks, as illustrated in the Figures 2.1(a) and 2.1(b). Features of a batch recipe include the amount to be processed by a discrete task as well as the duration of the task. Batch processes are generally used for the production of low volumes of a variety of high value-added products using limited resources; hence production scheduling is of great essence in batch production.

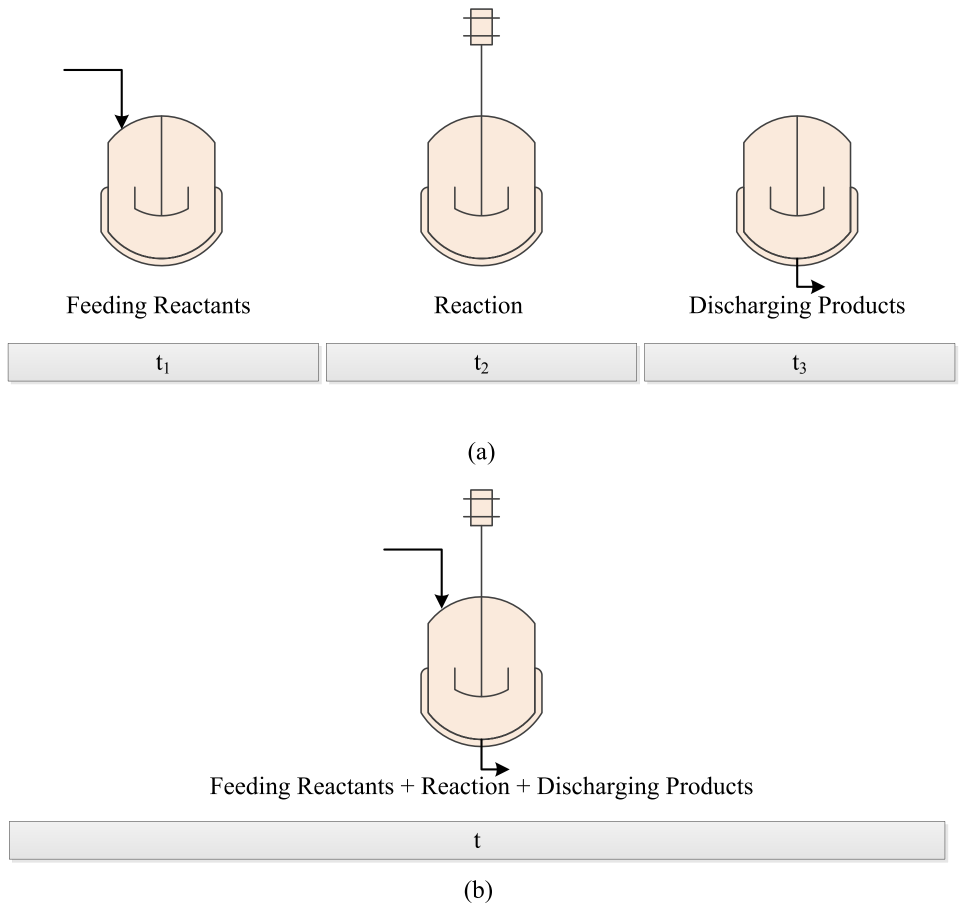
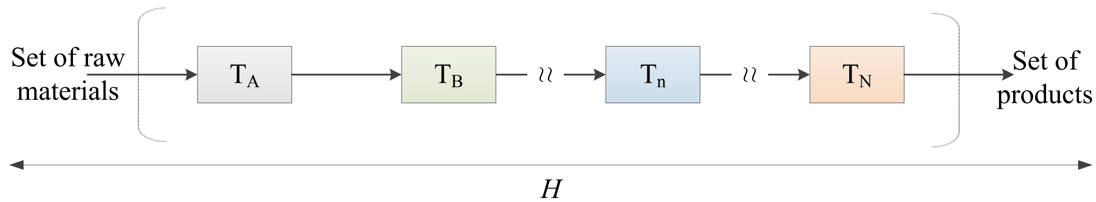
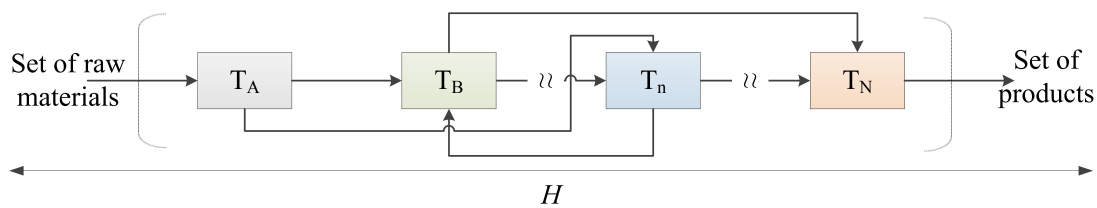


Figure 2.1 (a) Batch reactor (b) Continuous Reactor

Batch processes can be classified according to process layout into single and multiple stage processes. The sequence of stages that a batch process adopts is informed by the batch/product recipe. Each stage can have a single unit or multiple units operating in parallel. Multiple stage batch processes can be further classified into two categories; multiproduct and multipurpose. Multiproduct batch processes are appropriate for manufacturing products with identical and fixed recipes; see Figure 2.2(a). Multipurpose batch facilities are appropriate for the manufacturing of products characterized by a variation of production recipes as illustrated in Figure 2.2(b).



(a)



(b)

Figure 2.2 (a) Multiproduct batch process (b) Multipurpose batch process

The discrete nature of batch processes brings with it a feature that is easily suppressed in continuous processes, i.e. time. The capturing of this extra dimension is the reason why scheduling of batch processes is more complex. Methodologies designed for continuous processes cannot be directly applied to batch operations since they do not take into account the time dimension. Other challenges encountered when dealing with batch processes include product recipe representation, storage policies, changeover, etc. (Méndez et al., 2006).

2.3.1. Recipe representation

A production recipe of a batch operation has a significant influence when developing optimization models. The recipe presents the layout of the production line and includes information such as the sequence in which batches should be processed, mixing and splitting of operations, and material recycles. A recipe representation intends to describe the actual process of converting raw materials into desired products, unlike flowsheet representations that describe the actual plant. Different approaches for representing batch production recipes have been developed over the years. Kondili et al. (1993) proposed a State Task Network (STN) representation. As portrayed in Figure 2.3(b), two types of nodes are included in the STN representation. These are the state nodes (circular in shape), representing the feeds, intermediate and final products; and the task nodes (rectangular in shape), representing different operations that transforms feed/s into product/s. Directed arcs between nodes represent task precedence. The STN representations explicitly show all feedstocks sent to a task and all states produced by a task. Most mathematical formulations based on an STN representation have sets of states and tasks as indices.

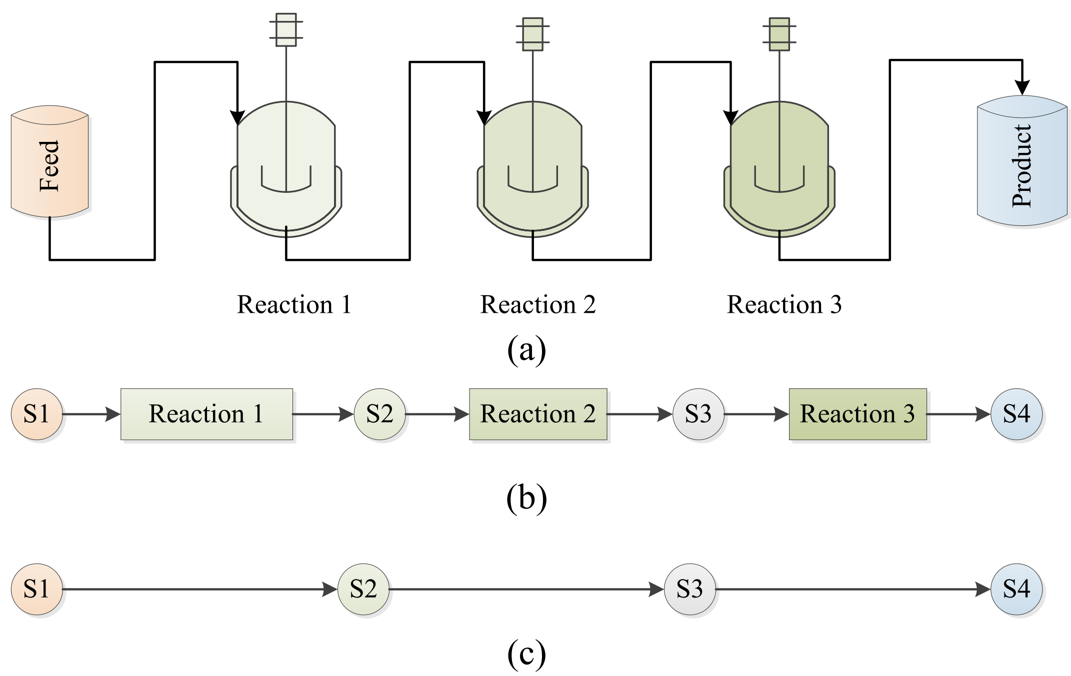


Figure 2.3 (a) Process flowsheet (b) STN (c) SSN

Pantelides (1994) proposed the Resource Task Network (RTN) representation. In addition to the STN, the RTN also includes utilities such as transportation, cleaning, etc. Types of resources in an RTN representation includes those that are consumed temporarily (e.g. units), those that are consumed or produced permanently (materials) and those with an availability profile (utilities). The RTN representations disaggregate tasks if multiple units are suitable. Most mathematical formulations based on an RTN representation have sets of resources or utilities and tasks as indices.

Smith (1996) proposed a representation that decomposes a process system into process materials and process equipment, the State Equipment Network (SEN). Equipment refers to physical devices that execute tasks. The construction of the SEN generally leads to a smaller combinatorial problem for the selection of equipment (Yeomans and Grossmann, 1999). For problems where every equipment is restricted to perform a single task, an SEN representation can be similar to an STN representation. In SEN representations, only one interconnection of state goes into an equipment and another one leaves the equipment, even when an equipment is suitable to process many tasks. The state definition is, therefore, not unique since properties of the streams will be determined by a particular task that the equipment performs. This means that the state definition will have to consider all the possible realizations of the streams that will originate from a certain task in an equipment, which can complicate the modeling stage (Yeomans and Grossmann, 1999).

The State Sequence Network (SSN) was introduced by Majozi and Zhu (2001). As displayed in Figure 2.3(c), the SSN representation only has the state nodes, and the task occurring in a unit is represented implicitly. For example, a heating or boiling task and a unit where this task occurs will be implicitly represented if a node representing water in a liquid phase is connected to a node representing water in a vapor phase. This approach was developed by realizing that the usage of a state corresponds to the existence of a task and the production of another state. Also, the capacity of a unit in which a particular state is used or produced sets an upper limit on the amount of state used or produced by the corresponding task. By noting these realizations, one state can be chosen and other states can be represented in terms of the chosen state. The chosen state is called the effective state and it should remain consistent throughout the formulation. Effective states are considered when defining binary variables. Therefore, the resulting number of binary variables becomes a product of the number of effective states involved in the process and the total number of time points used in the formulation. Task and unit binary variables are not required in SSN-based models as opposed to STN-based and RTN-based models.

2.3.2. Storage policies and wait times

The storage policies are classified according to the availability and capability of storage for storing final products and/or intermediate products in a batch process. In Finite Intermediate Storage (FIS) policy, intermediate products are stored in a storage tank of limited capacity. Unlike in FIS, the availability of storage for intermediate products is guaranteed in Unlimited Intermediate Storage (UIS) policy. Common Intermediate Storage (CIS) policy involves the sharing of storage tanks by various tasks within the plant. Washing of storage tanks is therefore required to avoid the contamination of products. FIS, UIS and CIS operational philosophies are illustrated in Figure 2.4. Sometimes an unused processing unit can be used to store final products and/or intermediate products and this is referred to as Process Intermediate Storage (PIS) operational philosophy. Mixed Intermediate Storage (MIS) policy is the one that includes a combination of two or more of the above-mentioned policies.

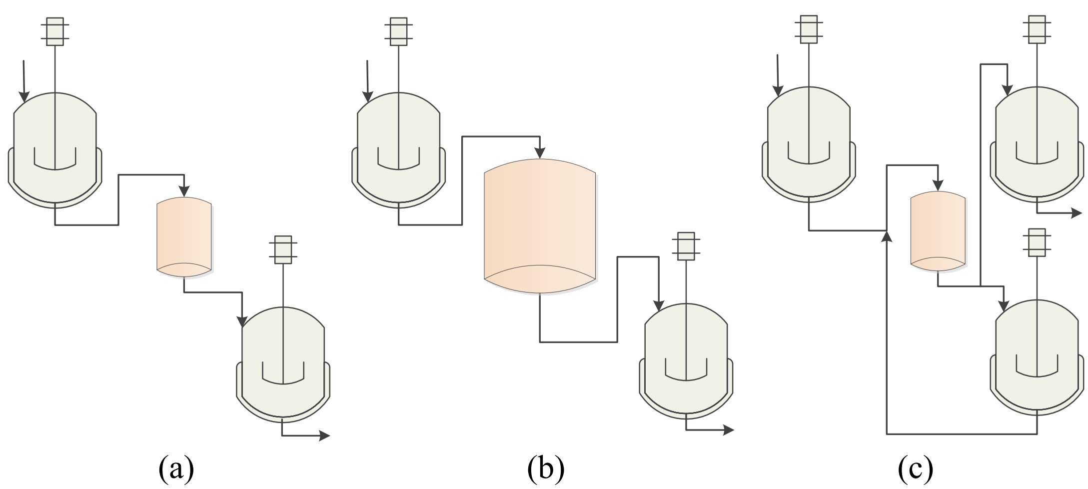


Figure 2.4 (a) FIS (b) UIS (c) CIS

Storage tanks occupy a significant area in facilities where the operational space is of essence. No Intermediate Storage (NIS) operational philosophy allows intermediate products to wait in the same units they were produced in (post-processing unit-wait times) and/or in the unit that will do the further processing (pre-processing unit-wait times) (Majozi et al., 2015). In post-processing unit-wait times, intermediate products are allowed to wait in a unit that produced them while waiting for the next unit to be ready for further processing. In pre-processing unit-wait times, a state is stored in a unit that will do further processing while waiting for other feed states, i.e. a task that requires more than one intermediate state, and this is called non-simultaneous material transfer. When dealing with unstable intermediate products that need to be sent to the next task as soon as they are formed; the Zero Wait (ZW) policy is adopted. Less sensitive intermediate products can be allowed to wait for a limited period of time under the Finite Wait (FW) policy. Highly stable intermediate products can be allowed to wait for a long period of time under a policy called the Unlimited Wait (UW).

2.3.3. The time dimension

The nature of batch processes require optimization models to take time into consideration since discrete tasks are processed at different times across the time horizon of interest. In the early stages of development of this research area, handling time when modeling batch processes were through Time Average Models (TAMs). This approach fails to truly represent batch processes since it treats batch operations as pseudo-continuous operations (Majozi, 2010). Another approach involves treating time as a known fixed parameter with no opportunity to change within the desired time horizon. This approach deprives the model of solving to true optima. An alternative approach would, therefore, be to allow time to be flexible and vary across the desired time horizon. This, however, brings with it another challenge of how the time horizon of interest is represented.

Based on how time is represented across the time horizon of interest, optimization models for batch processes can be classified into discrete and continuous-time formulations. The former evenly divides the time horizon of interest into a finite number of time intervals of known duration; see Figure 2.5(b). The starting and finishing times of tasks are then allowed to happen only at the boundaries of these intervals. Kondili et al. (1993) presented an MILP framework, based on the discretization of the time horizon into a finite number of equal intervals of known duration. The time horizon of interest was discretized into uniform time intervals that coincided with the beginning and/or end of a particular task. The inflexibility in the timing decisions generated infeasible and/or suboptimal production schedules. Also, the accuracy of discrete models increases with the number of time intervals. For some problems, for example those with duration of task that has decimals, the number of required intervals can be very large. A scheduling problem with a task that has a 4.2 hour duration will have many uniform time intervals of 0.2 hours. The large number of time intervals would result in an explosive binary dimension of the problem which will be computationally expensive to solve. Avoiding this by rounding off the duration of tasks with decimals into whole numbers, for example rounding 4.2 hours into 4 hours, will yield inaccurate results.

Shah et al. (1993) provided an examination of the computational issues encountered by Kondili et al. (1993). They proposed complementary measures of modifying both the formulation and the branch and bound solution procedure in order to reduce the computational time. Their technique included reformulating allocation constraints in order to tighten the LP relaxation of the MILP so that it can be solved within fewer LPs. They also examined ways in which the size of the relaxed LP can he reduced significantly by eliminating binary variables and a large proportion of the constraints from the LP relaxation of the MILP, thus resulting in a much smaller problem to be solved at each node of the branch-and-bound procedure. These measures are however specific for the solution of the resulting model. In cases where a reasonable number of intervals is sufficient to obtain the desired problem representation, optimization models based on discrete time representation have proven to be efficient, adaptable, and convenient for a range of industrial applications (Méndez et al., 2006).

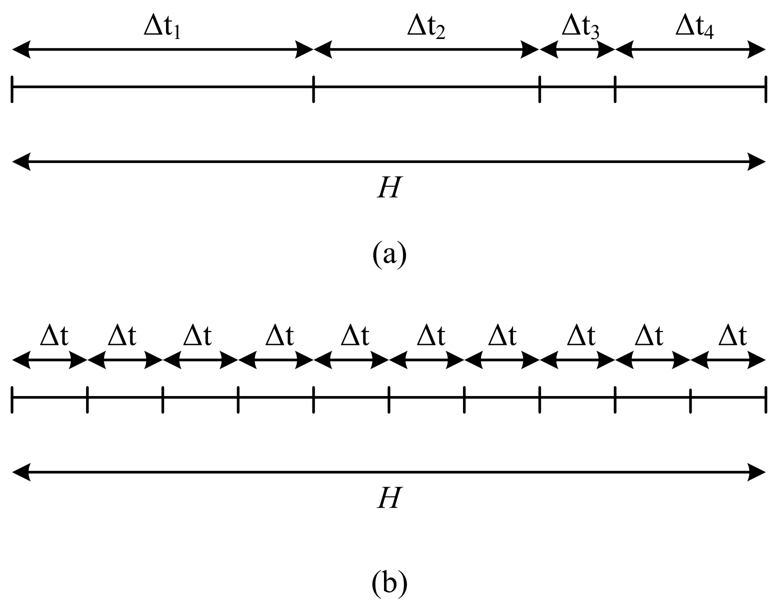


Figure 2.5 (a) Uneven and (b) Even time discretization

The drawbacks of discrete time formulations can be avoided by using continuous-time representation. In these formulations, the time horizon of interest is unevenly divided into a finite number of unknown intervals using variables that capture the exact time at which a task starts or finishes; see Figure 2.5(a). The number of variables is therefore significantly reduced and the flexible timing decisions can lead to feasible solutions.

Continuous-time formulations involve alternative event representations for network batch processes i.e. global and unit-specific event-based. Global event-based use unknown uniform events where the time associated with events is common across all units. In other words, the beginning and the finishing times of the set of batch tasks are linked to specific time points. In contrast to global time points, the time associated with the events can be different across all units in unit-specific representation. In other words, different tasks are allowed to start and/or finish at different times for the same event point. Formulations based on global time points or unit-specific time events strongly depend on the number of time or events points predefined. Since this number is unknown a priori, it can be determined through an iterative procedure where the number of time points or events is increased by 1 until there is no improvement in the objective function. Continuous-time formulations are generally more complex and have a higher integrality gap which indicates a poor approximation ratio. They have, however, proven to better capture the time dimension in batch processes.

2.3.4. Introduction to scheduling of batch processes

Scheduling refers to the allocation of resources to processing tasks over time. This includes determining what task to execute, where to process tasks, which sequence to follow, when to execute tasks, and sometimes a number of raw materials that should be processed in each task. This information is important when designing process operations and/or optimizing production performance. Production scheduling is very crucial in operating batch processes in a sustainable way, yet it is a challenging task especially in flexible batch facilities that allows the production of different products within the same facility (Floudas and Lin, 2004).

Traditionally, production scheduling was performed manually by trained personnel using practices recorded from previous experiences. Manual scheduling became extremely challenging due to increased production volumes, alternative production recipes, volatile production orders and the need to save energy, water and minimize other operating costs (Harjunkoski et al., 2014). The ideal way of considering the aforementioned and other factors when developing a profitable production schedule is through optimization. Optimization solutions achieve both economic and environmental benefits.

Scheduling models are based on concepts of arranging events of a schedule over time with the aim of guaranteeing that the maximum capacity of the shared resources is not exceeded. Types of production schedule formulations according to the considered time horizon are; short-term (in days), medium-term (in weeks), and long-term (in months) (Majozi et al., 2015). The corresponding models deal with the allocation of a set of limited resources over time to manufacture one or more products following a batch recipe (Méndez et al., 2006). The studies reported a wide range of scheduling problems that have been solved using different optimization approaches such as graphical techniques, mathematical modelling (LP, MILP, and MINLP), heuristic methods, artificial intelligence methods, and evolutionary algorithms. Most of these methods are often presented in literature from a purely modeling point of view and tested only on small-scale examples (Harjunkoski et al., 2014).

2.4. Recent continuous-time scheduling formulations

Excellent reviews on scheduling have been presented by various authors (Méndez et al., 2006; Floudas and Lin, 2004; Harjunkoski et al., 2014). Major challenges in the development of scheduling formulations include achieving global optimality, the reduction of binary variables and computational times.

Schilling and Pantelides (1996) presented a continuous-time scheduling formulation based on the RTN representation of Pantelides (1994). In their formulation, the overall scheduling time horizon was demarcated into time intervals of unknown lengths, and the boundaries of each time interval coincided with the start and/or finish of a particular task/s. A single binary variable was used to describe units (j) and tasks (i) at any point in time t, i.e. y_{ijt}.

Ierapetritou and Floudas (1998) applied the model of Schilling and Pantelides (1996) to a simple process where a single product is produced through three stages: mixing, reaction and separation. Given the simplicity of the process, the formulation of Schilling and Pantelides (1996) was observed to have a large number of constraints (220), continuous variables (157), binary variables (46) and intergrality gap (138%). Ierapetritou and Floudas (1998) presented a formulation that, when applied to the above example, had smaller number of constraints (108), continuous variables (105), binary variables (15) and intergrality gap (28%).

Ierapetritou and Floudas (1998) achieved the above results by introducing unit-specific event-based models. They proposed a continuous time formulation for short term scheduling of multipurpose batch processes based on the STN process representation of Kondili et al. (1993). In trying to avoid a large number of binary variables (with a dimension of $i \times j \times t$) which may result when a single binary variable y_{ijt} is used, Ierapetritou and Floudas (1998) separated units and task events by assigning corresponding binary variables v_{jn} and w_{in}, respectively. This lead to a much smaller number of binary variables for processes with several tasks and units. However this model initially predicts a large number of binary variables, in situations where stages involve several units, that can later be reduced by exploiting one-to-one correspondence between tasks and units. This reduction procedure can however be complicated for large problems.

To achieve the least number of binary variables without using the variable reduction procedure, Majozi and Zhu (2001) eliminated the need for task and unit binary variables by introducing the State Sequence Network (SSN). Only states are considered and a single variable y_{sp} is used throughout the formulation. Majozi and Zhu (2001) also introduced the aggregate model where the number of binary variables is reduced by treating multiple units in a stage as one. This can be done when the units involved in a particular stage have the same performance and when the process in a stage are operated in phase.

Janak et al. (2004) proposed an enhanced unit-specific event-based formulation for short-tern scheduling of multipurpose batch processes. Their work expanded on the work of Ierapetritou and Floudas (1998) by incorporating features such as storage policies (UIS, FIS, NIS, and ZW), resource constraints, variable batch sizes and processing times, batch mixing and splitting, and sequence-dependent changeover times. In their formulation, Janak et al. (2004) defined new tasks for the storage of states and the utilization of resources. They also introduced two binary variables, i.e. ws_{in} indicating whether or not a task starts at each event point and wf_{in} indicating whether or not a task ends at each event point.

Sundaramoorthy and Karimi (2005) argued that the de-coupling of a 3-index binary variable (y_{ijt}) into two 2-index binary variables (v_{jn} and w_{in}) does not reduce the overall number of binary variables as suggested by Ierapetritou and Floudas (1998). They demonstrated that decoupling of tasks from units increases the number of binaries by increasing the number of tasks, and at the same time decreases them by eliminating the v-variables, but the net effect of these two actions is zero on the number of binary variables. They added that the only difference between the 3-index y-variables and the 2-index w-variables is that the former display the unit information explicitly in terms of j, while the latter hide the same behind i. A formulation presented by Sundaramoorthy and Karimi (2005) is a slot-based continuous-time formulation that does not decouple tasks from units. When compared with unit-specific event-based models, however, their model gave suboptimal results and increased computational time.

Shaik et al. (2006) presented a comparative study where they assessed the performance of different continuous-time models when applied to several benchmark example problems in literature. The comparison was with respect to the problem size (in terms of the number of variables and constraints), computational times (on the same computer), and number of nodes taken to reach zero integrality gap. They concluded that unit-specific event based models require less events and they perform better than global event based models and slot-based models. This was because they observed that both the slot-based and global event-based models always require the same number of event points, while the unit-specific event-based models require less event points to solve a problem to global optimality. Due to heterogeneous locations of event points used, unit-specific event-based approach is considered the most general and most rigorous representation of time used in short-term scheduling models.

Janak and Floudas (2008) proposed a framework for reducing, and sometimes even closing, the integrality gap experienced by many complex unit-specific continuous-time formulations for short-term scheduling problems. Their methodology involve four steps: analysing the STN representation of the problem in order to determine its practical limitations (e.g. when tasks cannot take place and which unit will be the bottleneck of the process); considering new constraints (e.g. tightening constraints and bounds on the sums of key variables); solving supporting problems in order to get tighter values for the bounding constraints and to determine the minimum number of event points; and lastly introducing the reformulation-linearization technique (RLT) to provide tighter problem formulations. The RLT was developed by Sherali and Adams (1994) and it consists of a reformulation phase and a linearization phase. In the reformulation phase, selected constraints and binary variables are multiplied and the resulting new constraints are added to the original problem. Then the nonlinear model is then linearized during the linearization phase. Janak and Floudas (2008) argues that the addition of these new inequalities gives a higher dimensional representation of the feasible region for the problem and thus yields a tighter LP relaxation.

Janak and Floudas (2008) and Shaik and Floudas (2009) demonstrated that not allowing tasks to span over multiple event points might yield suboptimal solutions in some cases. Shaik and Floudas (2009) established that both the original model of Ierapetritou and Floudas (1998), and their improved models, may give suboptimal solutions because they do not allow tasks to occur over multiple events. Shaik and Floudas (2009) also established that the formulation of Janak et al. (2004) which was developed to address a more general has weak LP relaxation and requires a large number of constraints, nonzeros, and CPU time. To reduce the complexity and improve the efficiency of the model of Janak et al. (2004), Shaik and Floudas (2009) proposed a novel unified model that allows tasks to occur over multiple event points. Their model requires an extra set of iterations that control the number of event points that a task is allowed to span. When analyzing the limitations of unit-specific event-based models, Li et al. (2010) confirmed that the work of Shaik and Floudas (2009) indeed addressed the limitations of previous models by allowing a task to span several event points.

Susarla et al. (2010) presented models that use unit specific slots that allowed tasks to span over multiple slots and also allow non-simultaneous transfer of material into a unit to get a better schedule. This means that for a task that requires more than one intermediate materials, it is possible for some materials to be stored in a unit that is processing that task while waiting for the other intermediate materials. The model of Susarla et al. (2010), and all other unit-specific event-based models in literature at this stage, assumed unconditional sequencing. This means that different tasks in different units are always aligned without monitoring the actual material flows. These models assume that consumption tasks at event n + 1 are always aligned with production tasks at event n irrespective of whether the material produced from a production task is actually used or not.

Seid and Majozi (2012) introduced conditional sequencing where producing and consuming tasks of an intermediate state are aligned only when a consuming task actually uses the material from a producing task. Using the SSN recipe representation, Seid and Majozi (2012) presented a formulation where each task starts and finish at a particular unit specific slot. Their model requires less computational time to reach global optimality when compared to existing formulations in literature at this stage. However, Vooradi and Shaik (2013) argued that the model Seid and Majozi (2012) used partial conditional sequencing since it aligns a production task with all consumption tasks even if a single consumption task uses material from that production task.

The formulation of Vooradi and Shaik (2013) had rigorous conditional sequencing. This means that production and consumption tasks are aligned by accurately monitoring the material flow from each production task to each consumption task. When compared with partial conditional sequencing, rigorous conditional sequencing further reduces the number of events required. The scheduling formulation of Vooradi and Shaik (2013) can also effectively handle cases with non-simultaneous material transfer through proper handling of pre-processing and post-processing unit wait times.

2.5. Recent wastewater minimization formulations

Manufacturing industries contribute significantly to the generation of wastewater that pollute the environment. Industries responded to the wastewater problem, as outlined in chapter 1, by sending the generated effluent to treatment plants. Depending on the characteristics of wastewater, treatment methods are classified into physical, chemical and biological methods (Tchobanoglous et al., 2014). High capital investments are required for these treatment facilities, and the operating cost of the treatment operations depend on the amount of wastewater as well as the nature and the concentration of the contaminants. As a result, industries are trying to find techniques of minimizing the amount of wastewater as well as controlling the toxicity of the wastewater.

Most of the early research studies on water minimization were developed for continuous processes (Chwan and Foo, 2009). This was because continuous manufacturing processes generated larger volumes of wastewater and they were very popular. Batch processes, on the other hand, were less popular and more complex due to the existence of the time dimension. Batch processes have gained more attention due to the increased demand for various low-volume high-value-added products. Wastewater produced by batch processes is generally more toxic than the wastewater produced by continuous processes. Wastewater minimization techniques for continuous processes cannot directly apply to batch facilities due to the extra time dimension. Techniques for water minimization in batch processes have gained attention in the past decade. Techniques for minimizing water in batch plants are classified into graphical and mathematical programming.

Washing of equipment, when changing over from one task to the other in the same unit, is the major source of wastewater in most multipurpose batch facilities. In some batch operations, water is used as a medium for solvent extraction which is then dispensed at the end of the process. Minor sources of wastewater exist which may include floor washing. Wastewater generated in batch processes is mostly composed of toxic concentrations of contaminants. Techniques for water minimization in batch processes need to, therefore, satisfy both the contaminant concentration constraints as well as the time constraints, which makes them more complex than their continuous counterparts. Some models fix the outlet contaminant concentration and leave the amount of wastewater as a variable to be minimized; while other models fix the amount of wastewater and leave the contaminant concentration as a variable to be minimized (Majozi, 2005b).

Popular process integration methodologies for wastewater minimization include direct, indirect and regeneration reuse and recycle. Direct reuse is when an outlet stream from a washing operation becomes an inlet stream to a washing operation in a different unit. Direct recycle is when an outlet stream from a washing operation becomes an inlet stream to a washing operation in the same unit. Direct reuse and recycle are illustrated in Figure 2.6(a). Two requirements need to be satisfied for direct reuse and direct recycle to occur: the time requirement (the finishing and the starting times of the washing operations must coincide) and the contaminant concentration requirement (the outlet contaminant concentration of the outlet streams need to be less than the maximum allowable contaminant concentration in the inlet stream). Indirect reuse and indirect recycle attempts to relax the time requirement by allowing water to be stored before it can be reused or recycled; see Figure 2.6(b). Regeneration reuse and regeneration recycle relaxes the contaminant concentration requirement by allowing water to be treated before it can be reused or recycled; see Figure 2.6(c). Regeneration is achieved with the aid of a water treatment technology. Buabeng-Baidoo et al (2017) achieved 85% reduction of wastewater generation by exploring multiple water reuse opportunities, including regeneration reuse by means of a reverse osmosis membrane, in the cleaning in place process of a large scale milk continuous processing plant.

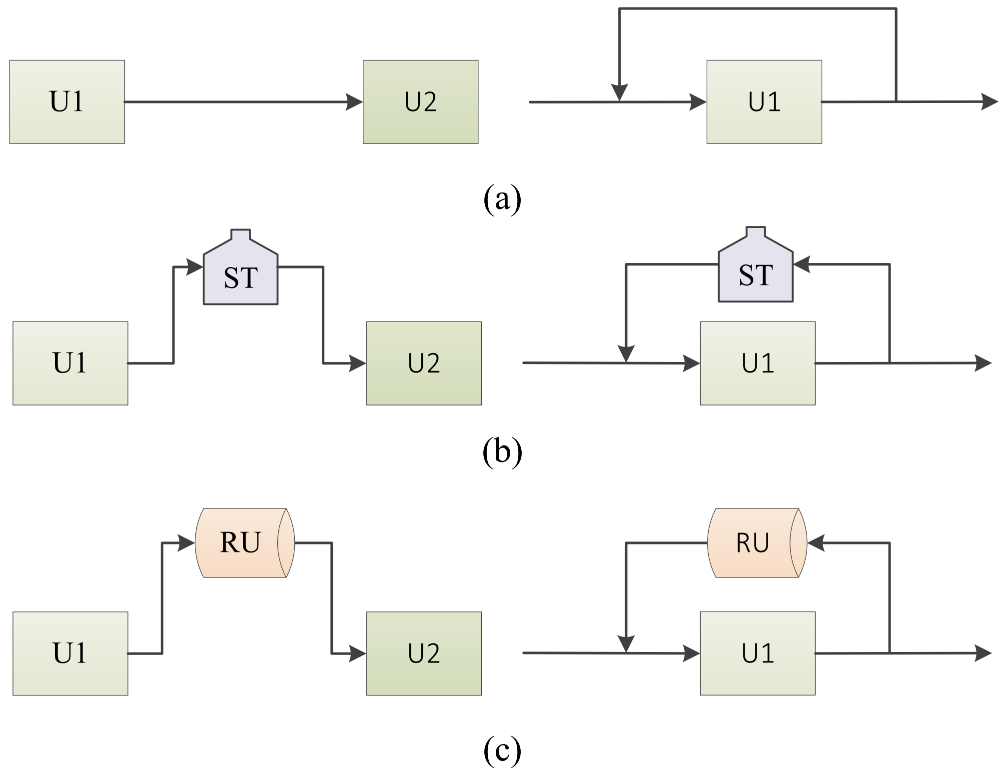


Figure 2.6 (a) Direct reuse and recycle (b) Indirect reuse and recycle (c) Regeneration reuse and recycle

Mathematical formulations are often based on a superstructure. The superstructure is presented as a diagram that represents all sources and sinks in a unified manner while considering all possible interconnections between various processes. The role of the optimization model is therefore to synthesize the best set of connections from the superstructure.

2.5.1. Insight based techniques

Wang and Smith (1994) presented a design methodology that aims to minimize water reuse between continuous water using operations. The first graphical technique for water minimization in batch processes, through the exploration of water reuse and recycle opportunities, was presented by Wang and Smith (1995b). Their targeting procedure includes dividing the problem into concentration intervals and time subintervals where the boundaries are set by the end-points of individual processes. Streams that are available for water reuse are then grouped in each time interval. Water available in each concentration interval is reused in the time subinterval. The surplus is reused in the subsequent time subintervals or stored for reuse in the subsequent concentration interval. Surplus water is neither allowed to be reused in lower concentration intervals nor in lower time subintervals. Freshwater is used after reuse opportunities are exhausted and the eventual surplus becomes effluent. However, the technique by Wang and Smith (1995b) demonstrated semi-batch behavior by allowing the reuse of water to occur between two units that are active. Majozi et al. (2006) improved on the work of Wang and Smith (1995b) and presented a graphical technique for water minimization in completely batch operations.

The technique by Majozi et al. (2006) is able to determine the water network and the minimum amount of freshwater that can be achieved by exploring reuse and recycle opportunities for strictly batch processes. The following information is required: the contaminant mass load, fixed water requirement, starting and finishing times of each batch operation, as well as the maximum inlet and outlet concentration. The issue of product mixing is however excluded since it is assumed that the considered processes are compatible and therefore product integrity is not compromised. Time is taken as a primary constraint. This technique recognizes that discrete amount of water is available either at the beginning and/or the end of the concentration or time interval. A hypothetical example can be used to illustrate the technique by Majozi et al. (2006).

The example involves the production of agrochemicals A, B, and C; in completely batch reactors. Sodium Chloride (NaCl) is formed in each of the three reactions as a byproduct and it is then removed through a liquid-liquid extraction product-washing stage where water is the aqueous phase. In the case of A, water is used solely for washing NaCl since the reaction took place in a solvent that is highly immiscible with water. In the case of B and C, water was used as a solvent and also for product washing. Table 2.1 summarizes the specification of the described problem. Duration of each task is given, together with the load of salt that should be removed, the amount of water required, as well as the contaminant concentration limits of water for different tasks ($C_{in,max}$ and $C_{out, max}$).

Table 2.1 Problem specification

Process	Time (h)	$C_{in,max}$ (Kg Salt/ Kg Water)	$C_{out,max}$ (Kg Salt/ Kg Water)	Water (Kg)	Salt load
A product washing	0.3	0	0.1	1000	100
Reaction B	0.4	0.25	0.51	280	72.8
B product washing	4.5.5	0.1	0.1	400	0
Reaction C	2.6	0.25	0.51	280	72.8
C product washing	6.7.5	0.1	0.1	400	0
Total				2360	245.6

Figure 2.7 provides the graphical representation of the example. Figure 2.7 also shows the concentration intervals: 0 to 0.1, 0.1, and 0.25 to 0.51 kg of salt per kg of water.

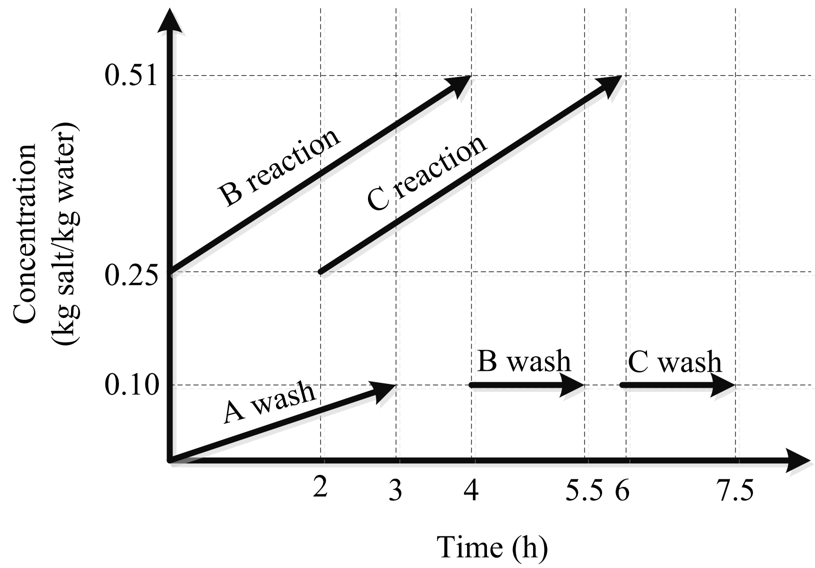


Figure 2.7 Graphical representation of the specified problem

Figure 2.8 shows targeting at the first concentration interval, 0 to 0.1, where washing of product A is the only operation. As presented in Table 2.1, 1000kg of water is required for operation A. The required amount will be freshwater since there is no reusable water available in this interval.

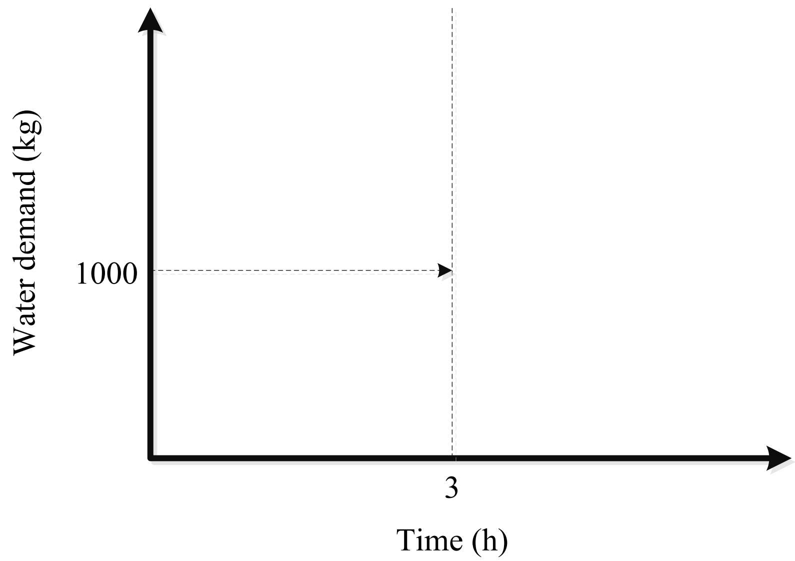


Figure 2.8 Targeting interval 0 to 0.1 kg of salt per kg of water

Figure 2.9 presents targeting at the concentration boundary of 0.1 kg of salt per kg of water. Productions B and C lie in this boundary as the concentration of water remains constant since no load is removed from the products. According to the Table 2.1, the combined water demand at this concentration boundary is 800kg. It is however evident from Figure 2.7 that both B and C starts after the completion of A wash. The outlet concentration from A wash corresponds to the required boundary concentration of 0.1. Waste from A wash can, therefore, be reused in B and C since both the time and contaminant concentration requirements for water reuse in batch processes are met. A water storage tank is needed to store water from the A wash since there is a time gap between the end of the A wash and the start of B and C wash.

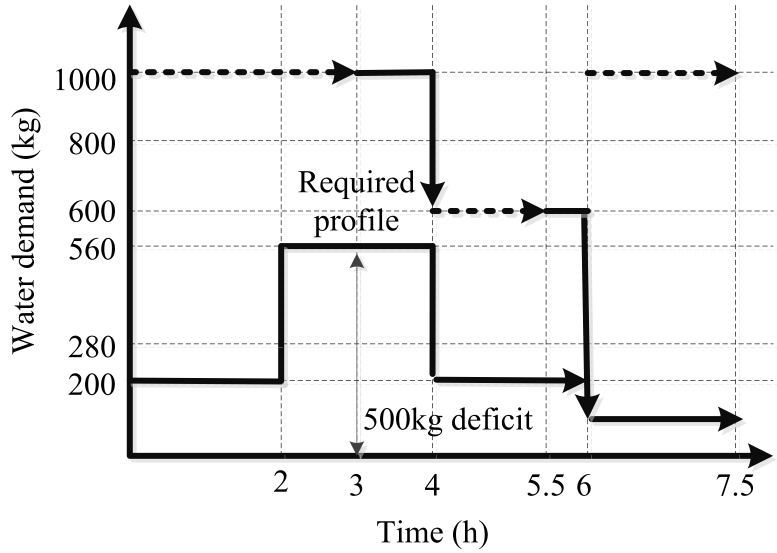


Figure 2.9 Targeting boundary 0.1 kg of salt per kg of water

Figure 2.10 represents the targeting at the interval 0.25 to 0.51 kg of salt per kg of water. This interval has the B and C reactions with the overall demand of 560kg as illustrated in figure 2.7. There is no reusable water available for these reactions since they both start before the completion time of A wash, hence freshwater is required.

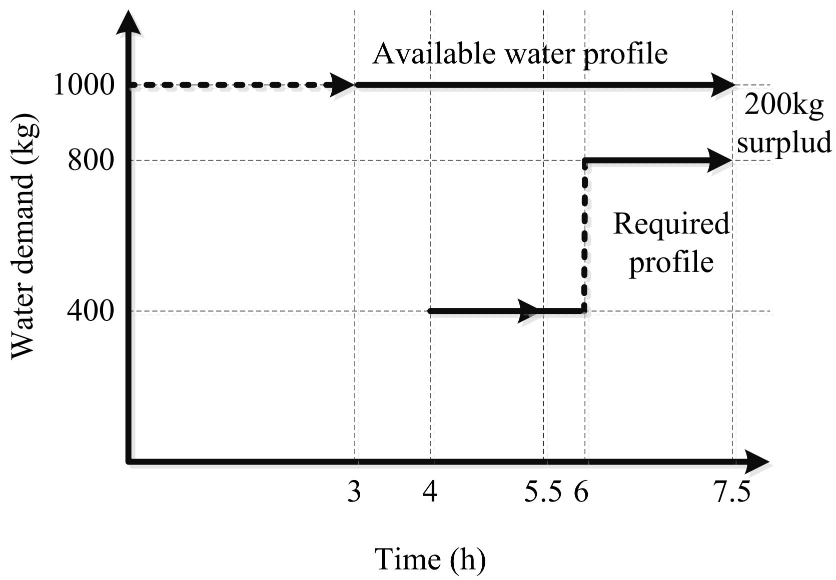


Figure 2.10 Targeting interval 0.25 to 0.51 kg of salt per kg of water

In this example, the total freshwater demand is 1560kg. This implies that 34% of freshwater was saved when using the proposed graphical technique. The work of Majozi et al. (2006) accommodated for completely batch operations by ensuring that water is only available or required at the end or the beginning of intervals and in discrete amounts.

Insight based techniques for water minimization provide insights by determining minimum freshwater targets and are useful when the time is treated as a fixed parameter in batch processes. Additionally, they are also limited to single contaminant problems. The aforementioned drawbacks can be overcome by using mathematical modeling techniques since they can address complex batch problems. Mathematical techniques for water minimization in batch processes are classified into those that are based on a predefined fixed schedule and those based on the variable schedule. Time is treated as a parameter on the former and as a variable on the latter.

2.5.2. Fixed schedule mathematical programming

Formulations for water minimization where the schedule is predefined are regarded as simpler to solve compared to those that determine the schedule as part of the algorithm. Almató et al. (1997) proposed a mathematical model for water minimization in batch processes, based on a predefined fixed schedule. They explored indirect water reuse opportunities by allocating storage tanks for reusable water. This means that water could be stored and used by a task that occurs at a later time. Direct water reuse opportunities were, however, not explored. Kim and Smith (2004) proposed a model that explored both direct and indirect water reuse opportunities where each unit producing wastewater was allocated a storage tank to avoid mixing. They urged that allowing wastewater mixing reduce opportunities for reuse due to higher contaminant concentrations.

A model by Majozi (2005a) explored direct water reuse for a fixed outlet concentration scenario. The formulation included sequencing constraints that ensure that the starting time for the water using unit coincide with the finishing time of the water producing unit for direct reuse to occur. Bilinear terms, comprising of continuous and binary variables, were linearized exactly using Glover transformation (Glover, 1975) and the resulting model was MILP. A formulation proposed by Li and Chang (2006) determines the number and sizes of storage tanks, the configuration of pipeline network as well as the operating policies of water flows. Buffer tanks were incorporated to provide opportunities for indirect water reuse and to equalize the flow and concentration of wastewater before entering the treatment systems. Chen et. al (2008) analyzed the impact of central storage facilities on freshwater reduction and their model synthesizes water-using networks with the minimum freshwater consumption. Recently, Lee et al. (2014) presented a fixed schedule model that simultaneously targets minimum water and wastewater flow, storage capacity, and interconnections for multi-contaminant cyclic batch operations. Their formulation is capable of identifying the water source or sink to be reduced or eliminated, predicting the amount of external water required, water source to be reused, recycled, regenerated, or discharged and determine the minimum storage tank capacity and interconnection configurations.

Assuming an optimum fixed schedule limits the water minimization model from finding more water reuse/recycle opportunities. For example, if the finishing time of a water-producing operation does not coincide with the starting time of the water-using operation and they are both fixed, direct reuse will not happen. Whereas if time was treated as a variable, the water minimization model might have shifted the operations within the time horizon of interest, such that the finishing and the starting times of the operations coincide and the direct water reuse opportunity is explored.

2.5.3. Variable schedule mathematical programming

Mathematical techniques for water minimization based on an optimization scheduling platform can be further classified into discrete and continuous-time formulations. Cheng and Chang (2007) presented a discrete time formulation that simultaneously optimizes the schedule, water reuse opportunities and wastewater treatment by incorporating all three optimization problems in one platform. The nature of discrete models results in large model sizes that require more time to solve hence continuous time models are preferred.

Majozi (2005b) presented a continuous time variable schedule mathematical model for wastewater minimization in batch processes, built on a scheduling platform presented by Majozi and Zhu (2001) which is based on a State Sequence Network (SSN). Their model explored four scenarios: fixed outlet concentration without reusable water storage, fixed water quantity without reusable water storage, fixed water concentration with reusable water storage, and fixed water quantity with reusable water storage. The first two scenarios explored direct reuse and recycle opportunities and the last two scenarios explored indirect reuse and recycle using a central reusable water storage tank. Majozi and Gouws (2009) presented a model that explores direct and indirect water reuse and recycle with central reusable water storage tank for multi-contaminant problems.

Water can also be minimized between multiple processing plants that are grouped in different geographical locations through interplant water integration. Chew et al. (2008) explored direct and indirect interplant water integration through pipelines and centralized utility hub. Optimization approaches for water minimization, insight-based and mathematical optimization, can be combined and this provides the opportunity to use targets obtained beforehand to generate alternative networks (Oliver et al., 2008). Gouws et al. (2010) reviewed earlier formulated water minimization models.

A method presented by Li et al. (2010) simultaneously optimized production and water network. They incorporated regeneration that reduces the contaminant concentration of wastewater in order to improve indirect reuse opportunities. Adekola and Majozi (2011) expanded on the work of Majozi and Gouws (2009) by incorporating a black box regeneration unit that treats water and increases reuse and recycle opportunities. A model by Chen et al. (2011) simultaneously optimized the production schedule and the water network for periodic operations. Their model was built on an RTN scheduling framework of Chen and Chang (2009). Nonyane and Majozi (2012) presented a variable schedule model for water minimization that can handle longer time horizons.

Grundemann et al. (2012) conducted an experimental investigation aimed at reducing cleaning related wastes, including wastewater, by transferring macro batch to micro continuous campaign manufacturing. Their three-step approach to design and optimizing a micro-continuous process starts by exploring how fouling and deposits can be avoided by choice of equipment. The frequency of cleaning is then minimized by exploring how batch production can be transferred to micro-continuous production through equipment dedication and proper production scheduling. They also argue that increasing the batch size reduces product frequency in a sequence which leads to fewer changeover procedures and less cleaning waste. The last step focuses on the optimization of the cleaning cycle by taking advantage of the small hold-up of the micro-continuous plant.

Variable schedule optimization models that incorporate heat and water minimization in one unified framework with an optimization scheduling model are also presented in literature (Seid and Majozi, 2014). In the work of Halim and Srinivasan (2011), the optimization problem is decomposed into the scheduling part, the heat integration part and the water reuse optimization part. The optimization problem was then solved sequentially starting with scheduling. However, the unified approach of optimizing resources simultaneously give better economic performance when compared to the sequential approach. Recent advances include a formulation by Chaturvedi and Bandyopadhyay (2014) that uses multi-objective functions for simultaneously minimizing freshwater requirement and maximizing production. The formulation was solved by repeatedly optimizing one objective while fixing others.

2.6. Handling changeovers

In batch processing, the changeover is a process of converting a unit or a production-line from processing one task to another. Changeover operations occur between tasks in a unit; for example washing, sterilization, equipment set-up, material transfer, etc. Research on changeovers was incentivized by two main factors: the loss of valuable production time since a unit is not operational during the changeover; and the cost of the changeover activities. Changeover can either be sequence dependent or sequence independent. Changeover time and/or cost will depend on both the task just completed and the task that is about to be processed if the changeover is sequence dependent. When the changeover is sequence independent, changeover time and/or cost is not influenced by the sequence of tasks in a processing unit. Sequence-independent changeovers are often found in situations where batches being processed have similar equipment set-up, operating conditions, etc.; and they can be easily modeled. For example, Li and Floudas (2010) incorporated sequence independent changeover or setup times into their scheduling model by lumping them into the processing time of batches.

Due to the common time grid for units, discrete time formulations can easily incorporate sequence dependent changeovers (Doganis and Sarimveis, 2007). Discrete-time formulations are not desirable to account for changeovers due to small changeover times that require finer time discretization leading to large model sizes and excessive computational times. Cerda et al. (1997) and Méndez et al. (2001) used a concept of order of precedence to handle changeovers in single stage multiproduct batch plants. Precedence relationships can either be global or immediate. However, precedence based formulations are not time grid-based and are suited for multiproduct batch plants that follow a linear process.

This review will explore continuous-time formulations that address sequence dependent changeover time and/or cost in multipurpose batch processes. Continuous time grid-based models overcome the drawback of discrete time formulations and are also suited for multiproduct batch processes. However, continuous time models require iterations when deciding the number of event points or slots and may result in higher integrality gaps.

Maravelias and Grossmann (2003) presented a continuous time scheduling model for multipurpose batch processes that incorporates sequence dependent changeover time into sequencing constraints of different tasks in the same unit. Their model used an STN representation and it made use of global time points. Janak et al. (2004) presented a similar scheduling model which made use of unit specific event based presentation which proved to be computationally better than the formulation by Maravelias and Grossmann (2003). Both these models explore changeovers of tasks that occur between consecutive time slots and place the unused time slots as the last slots in the time period. A model by Shaik and Floudas (2008) allows for changeover even in situations where the consecutive task does not occur in the next time slot, as long as there is no other task/s processed between the tasks involved in the changeover. In their respective work, Shaik and Floudas (2008) and Shaik and Vooradi (2013) handled sequence time by using equation 2.1.

$$T^{s}(i,n) \geq T^{f}(i',n') + tc_{i'i} - M\left(2 - \sum_{n''\in N} w(i,n,n'') - \sum_{n''\in N} w(i',n'',n')\right) - M\sum_{i''\in I} w(i'',n'',n''') \qquad 2.1$$

The above literature did not consider changeover cost. The requirement of accounting for sequence dependent changeover cost is the ability to determine a task that immediately follows the task that has just occurred in a unit. Erdirik-Dogan and Grossmann (2008) accounted for changeover cost in their scheduling model of multiproduct batch processes by introducing a binary variable $Z_{i,k,m,l,t}$. This binary variable, as defined by equation 2.2 to 2.4, becomes active when product i, assigned to slot l, is followed by product i' at slot $i+1$ on unit m at time period t. Their model places empty or unused slots as last slots in the time period.

$$z(i,k,m,l,t) \geq w(i,m,l,t) + w(i,m,l+1,t) - 1 \qquad 2.2$$

$$w(i,m,l,t) \geq z(i,k,m,l,t) \qquad 2.3$$

$$w(i,m,l+1,t) \geq z(i,k,m,l,t) \qquad 2.4$$

Kabra et al. (2013) accounted for changeover cost on their short-term model for multistage multiproduct batch process by introducing a binary variable $wc_{i,s',s,n'}$. This binary variable, as defined by equation 2.5 to 2.7, becomes active when state s' at event n' is followed by state s at event n provided that there is no other task occurring between n' and n. The formulation allows for empty events to exist between consecutive tasks. Changeover constraints of Kabra et al. (2013) are adapted from Shaik et al. (2009) for continuous processes.

$$wc(i,s',s,n') \leq w(i,s',n') \qquad 2.5$$

$$wc(i,s',s,n') \leq w(i,s,n) + \left(1 - \sum_{s'' \in Si} w(i,s'',n)\right) + \sum_{s''} \sum_{n' < n'' < n} w(i,s'',n'') \quad 2.6$$

$$wc(i,s',s,n') \geq w(i,s,n) + w(i,s',n') - 1 - \sum_{s''} \sum_{n' < n'' < n} w(i,s'',n'') \quad 2.7$$

Washing operations during changeover are inevitable in multipurpose batch processes. Adekola and Majozi (2017) presented a formulation that achieves wastewater minimization by exploring the sequence of tasks in a unit. They proposed a three index changeover binary variable $x_{ch}(s_{in,j}, s'_{in,j}, p)$ as well as a binary variable $XL(s_{in,j}, p)$ which becomes 1 when a task is the last task to be processed in a unit. The changeover binary variable $x_{ch}(s_{in,j}, s'_{in,j}, p)$ becomes 1 when a task $s_{in,j}$ at p is followed by $s'_{in,j}$ at a later time point since the formulation allows for empty time point between consecutive tasks in a unit. Equation 2.8 to 2.10 shows the relationship between the changeover binary variable $x_{ch}(s_{in,j}, s'_{in,j}, p)$ and the binary variable associated with the activeness of a task $y(s_{in,j}, p)$.

$$x_{ch}\left(s_{in,j}, s'_{in,j}, p\right) \leq y\left(s_{in,j}, p\right) \quad 2.8$$

$$x_{ch}\left(s_{in,j}, s'_{in,j}, p'\right) \leq y\left(s'_{in,j}, p\right) + \sum_{p' < p'' < p} \sum_{s''_{in,j} \in S_{in,j}} y\left(s''_{in,j}, p''\right) \quad 2.9$$

$$x_{ch}\left(s_{in,j}, s'_{in,j}, p'\right) \geq y\left(s_{in,j}, p'\right) + y\left(s'_{in,j}, p\right) - 1 - \sum_{p' < p'' < p} \sum_{s''_{in,j} \in S_{in,j}} y\left(s''_{in,j}, p''\right) \quad 2.10$$

$$\sum_{s'_{in,j} \in S_{in,j}} x_{ch}\left(s_{in,j}, s'_{in,j}, p\right) \leq 1 \quad 2.11$$

$$\sum_{s'_{in,j} \in S_{in,j}} x_{ch}\left(s_{in,j}, s'_{in,j}, p\right) + XL\left(s_{in,j}, p\right) = y\left(s_{in,j}, p\right) \qquad 2.12$$

$$\sum_{p} \sum_{s_{in,j} \in S_{in,j}} XL\left(s_{in,j}, p\right) \leq 1 \qquad 2.13$$

Constraints 2.11 and 2.13 ensures that there can only be one immediate successor to the current task and only one last task in a unit. Constraint 2.12 states that if a task occurs in a unit, it can either be followed by another task or it is the last task in that unit.

2.7. Background to mathematical modeling and optimization

Mathematical optimization is an approach that seeks to find the best solutions for problems defined mathematically through mathematical modeling. In the process industry, mathematical optimization can be done to minimize the total cost of design, optimize the operation (i.e. minimize operating cost and maximize profit), improve plant performance (i.e. yield, selectivity, use of resources, etc.) or improve environmental performance. Mathematical optimization problems consist of a process model and at least one objective function. In production scheduling, the objective can be to maximize or minimize the makespan, earliness, profits, inventory, cost, etc. A feasible solution to an optimization problem is defined as a set of variables that satisfy the constraint of an optimization problem. An optimum solution is the one that has the best objective function amongst the feasible solution in a feasible region.

A process model is a representation built to purposefully exhibit features and characteristics of an object, process or system. An optimization model consists of design variables that are involved in the trade-off. An optimum value of the design variable is desired since changing it may bring a benefit to one part of the design but a misfortune on the other. The objective of optimization is to find the values of these variables that yield the optimal value of the objective function.

Interrelationships of variables are captured using mathematical constraints. Constraints can be expressed as equalities or inequalities. A combination of constraints forms a model that can be optimized if an objective function is included. Models with 100, 400, and 1 000 000 constraints are considered as small, medium and large-scale problems respectively. Factors that make an optimization problem difficult to solve include the size of the model, types of variables and the nature of nonlinearity.

Types of mathematical models include empirical, stochastic and deterministic models (Dym, 2004). Empirical models attempt to describe the behavior of acquired data. Stochastic models are inherently random, i.e. similar parameters and initial conditions can lead to different outputs; whereas the output in deterministic models, which are based on the dynamics of the system, is fully determined by the parameters and the initial conditions. In operation research, deterministic models are used as process models where the system or process is described using mathematical equations, inequalities, and logical expressions.

The degree of freedom is determined as the difference between the number of independent variables and the number of constraints in a model. The degree of freedom of an optimization problem must be at least 1, meaning that there must be at least one variable which is free to vary. The problem is a uniquely solvable simulation problem when the degree of freedom is zero. The problem is over-specified when there are more independent constraints than variables and some constraints are therefore redundant.

Models can be built to describe the result of an observed system, to explain the behavior of a system, and/or to predict future behaviors. In many fields, such as engineering design, predictions by a validated and/or verified model influence decision making. Cobelli and Carson (2001) highlighted critical questions that are useful in general problem solving and also guides the process of building mathematical models, see Figure 2.11. In order to build a model that best predicts the desired outcome, it is essential to be clear of what is already known (parameters) and the assumptions that can be made. Before model predictions can be trusted, it is important that the model is verified or the outcomes are validated. Though the two can easily be confused, verification is different from validation. Model verification is the process of confirming that the model accurately represents the conceptual description of the system whereas model validation is done to ensure that the predictions of the model represent the real-life cases.

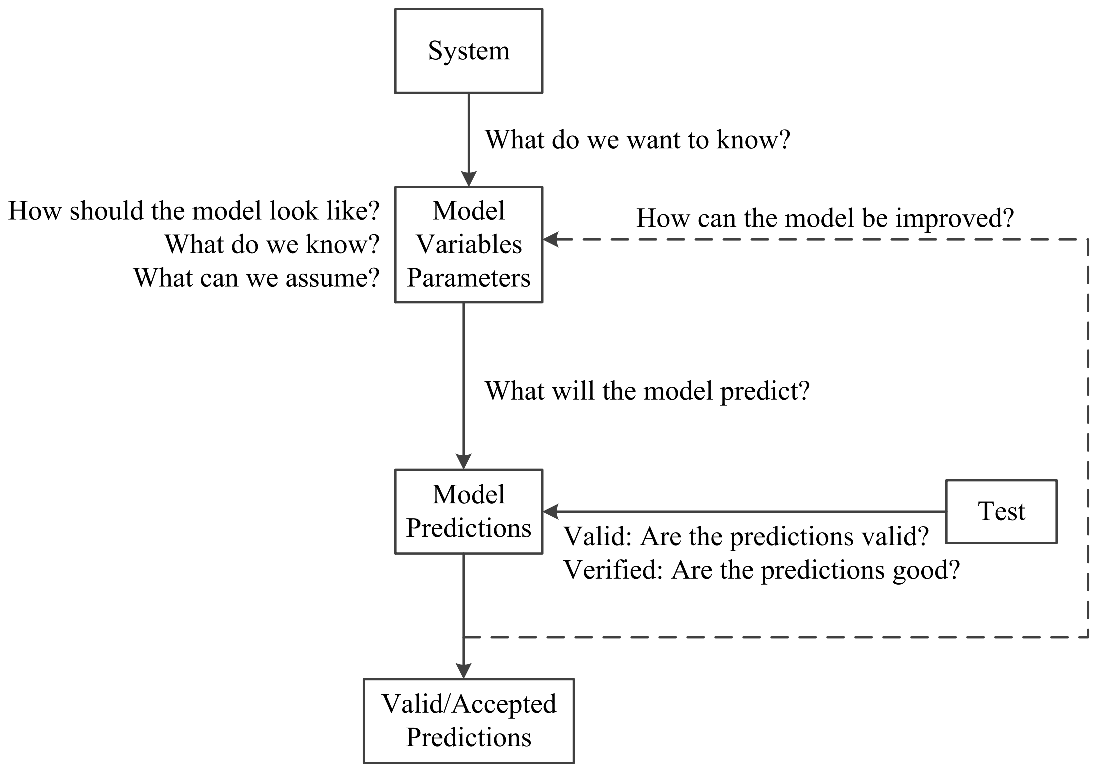


Figure 2.11 An overview of the process of developing models

2.7.1. Model classification

Variables used to build mathematical models can be continuous, discrete or binary. Continuous variables can take any value that is within the specified boundaries. Discrete, also known integer variables, can only take discrete values e.g. the number of storage tanks. Binary variables can either take a value of 0 or 1 indicating, for instance, the availability or unavailability of a processing unit at a particular time.

A mathematical model in which all the expressions in the process model and the objective function are linear is referred to as a Linear Programming (LP) model. Edgar and Himmelblau (1989) describe a linear expression as the one whose independent variables or derivatives appear only to the first power; otherwise, they are nonlinear. If at least one expression is nonlinear, that model is called a Nonlinear Programming (NLP) model. LP models are, in order of magnitude, easier to manipulate and solve than NLP models, hence there are techniques of linearizing NLP models (Glover, 1975). Applications of optimization models might require some variables to be whole numbers, integer variables. LP models consisting of a mixture of integer and continuous variables are called Mixed Integer Linear Programming (MILP) models (Williams, 1999). The NLP equivalent models are called Mixed Integer Nonlinear Programming (MINLP) models. MILP models guarantee global optimality and can be solved using the branch and bound technique.

2.7.2. Global optimization methods

Global optimization algorithms are used to solve mathematical optimization problems and can be classified as either stochastic or deterministic. Stochastic approaches are based on probability and often rely on physical analogues to guide the algorithm towards the global optimum solution. Stochastic approaches are not rigorous and have difficulty handling complex constrained problems. Deterministic algorithms may guarantee finite convergence, within a specified level of accuracy, by taking advantage of the mathematical structure of the optimization problem (Ryoo and Sahinidis, 1996). When building mathematical models for manufacturing industries, MINLP models usually surface. The following is a basic form of an MINLP problem:

Objective: Minimize $Z = f(x,y)$

Subject to: $g_j(x,y) \leq 0 \; j \in J$

$X \in X, y \in Y$

Where x and y are continuous and discrete variables, respectively. Major deterministic global optimization algorithms that can be used to solve MINLP problems include branch and bound, branch and reduce, general benders decomposition and outer approximation.

Branch and bound algorithm

Branch and bound algorithms are able to develop upper and lower bounds of the optimum objective value in sub-regions within the feasible region. This algorithm relaxes the discrete variables which then lead to a continuous NLP problem. The solution of the NLP at the node becomes the lower bound for the optimal MINLP objective function value which can be used to expand the nodes. Nodes can either be expanded breadth-first or depth-first. The breadth-first approach selects a node with the best value at each level and expands on all its successor nodes while the depth-first approach performs branching on the most recently created node within the tree. Branching occurs when the feasible region is being subdivided and bounding is the estimation of the upper and lower bounds of the global optimum solution. According to Ryoo and Sahinidis (1996), the depth-first approach requires less storage and can find the optimal solution early in the procedure.

The performance of bounding at every node in the branch and bound algorithm can be improved by pre-processing a global optimization problem using reduction techniques (Sahinidis, 2000). The method allows some nodes to be excluded based on the optimality and feasibility criteria. The resulting algorithm is called the branch and reduce algorithm. BARON (Branch and Reduce Optimization Navigator) solver make use of the branch and reduce method extended to continuous and discrete variables.

Generalized benders decomposition and outer approximation

Generalized benders decomposition and outer approximation algorithms differ from branch and bounds methods in that, for each major iteration, they solve an NLP problem (when all discrete variables are fixed) and MILP master problem. The NLP sub-problem provides the upper bound to the MINLP solution while the MILP master problem predicts both the lower bound to the MINLP solution and the values for the discrete variables for each major iteration. As the cycle of major iterations proceeds, the predicted lower bounds would increase and the search will be terminated when the lower bound coincide with the upper bound.

The general benders decomposition methods differ from the outer approximation methods on how they define their respective NILP master problems. The generalized benders decomposition method uses the optimal dual information to ensure that the master program corresponds to an initially poorly constrained integer linear program while the outer approximation algorithm uses the optimal primal information of the sub-problems to define a mixed-integer linear master program (Duran and Grossmann, 1986).

2.7.3. Convexity

The output of mathematical optimization can be a global optimum, the best solution, or a local optimum, one of the best solutions. Unimodal functions have one extremum which is a global minimum or maximum. Multimodal functions have multiple extrema where the smallest is the global minimum, and biggest is the global maximum, and the rest are local extrema. Whether the solution is local or global minimum or maximum can also be influenced by convexity. A function is convex if a line segment between any two points on the graph lies above or on the graph and concave if the line segment lies below the graph, see Figure 2.12. Strictly convex and concave functions have the line segment respectively above and below, and never on the graph. As can be observed from Figure 2.12, strictly convex or concave functions provide a single optimum. A global optimum solution can, therefore, be guaranteed for a convex function and not for a nonconvex function which may have multiple local optimum solutions (Lundell and Westerlund, 2012).

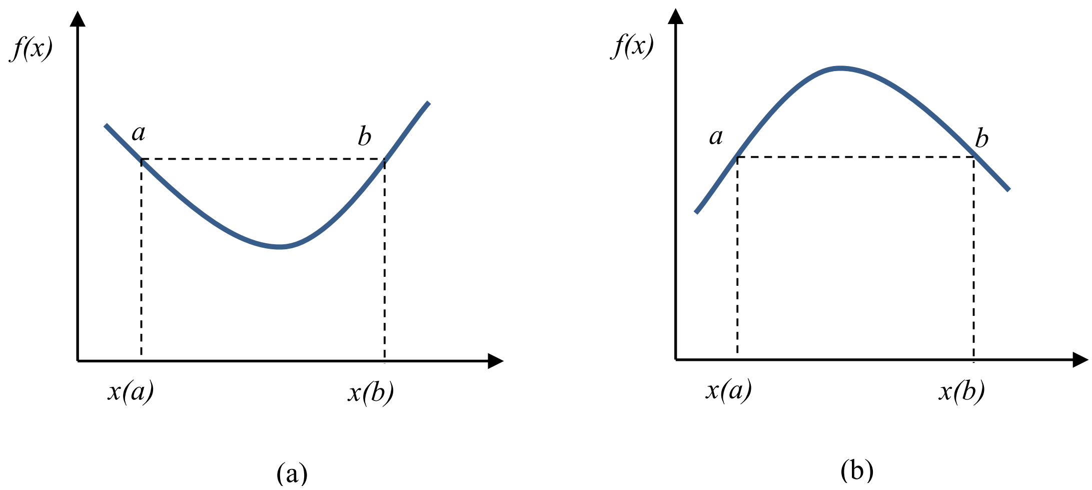


Figure 2.12 (a) Convex function (b) Concave function

For a one-dimensional function, convexity can be proven by finding the second derivative. A function is strictly convex or strictly concave if the second derivative is strictly greater or lesser than zero respectively. If the second derivative is greater or equal to zero, however, the function is convex though not strictly convex and if the second derivative is less or equal to zero, the function is concave though not strictly concave. To prove convexity for multivariable functions, a Hessian matrix is used to represent the second derivative and conditions similar to the ones mentioned above apply. However, there are convenient tests that can be made to establish the status of a Hessian matrix for strict convexity: all eigenvalues of the Hessian matrix must be positive, and all diagonal elements must be positive. For strict concavity: all eigenvalues of the Hessian matrix must be negative, and all diagonal elements must be negative.

MINLP formulations should, therefore, be convexified for the global optimum solution to be obtained. Figures 2.13(a) and 2.13(b) are illustrating envelopes for convex underestimators and concave overestimators to nonconvex and nonconcave functions respectively.

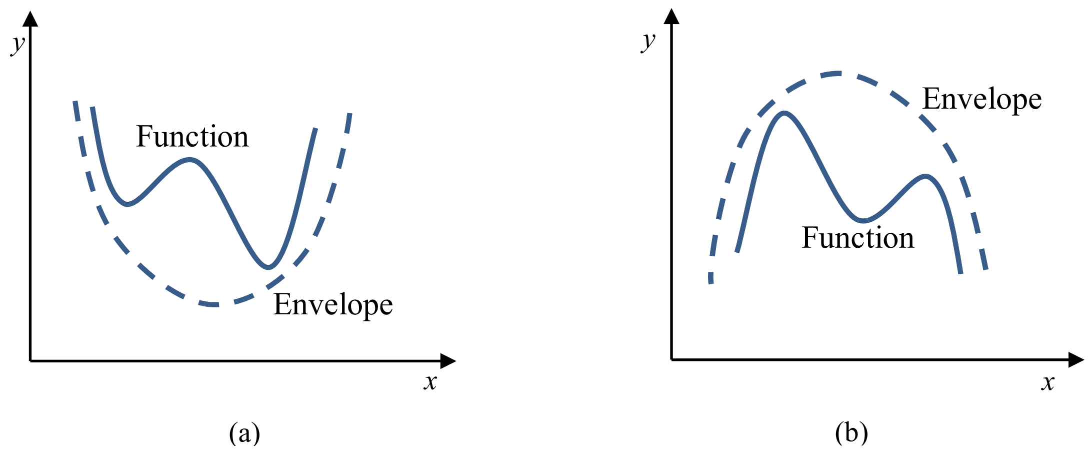


Figure 2.13 (a) Convex (b) Concave envelope

The following subsection will explore how nonlinearity caused by different bilinear terms can be transformed to linearity. This is important since nonlinearity causes models to be nonconvex which make it difficult to obtain and prove global optimality.

2.7.4. Linearization of bilinear terms

Bilinear terms can be caused by a product of variables. In this section, the following combinations will be discussed: product of two continuous variables, product of a continuous and a discrete variable, and a product of two discrete variables.

Product of two continuous variables

A method proposed by Mccormick (1976) can be used to linearize a bilinear term of a product of two continuous variables by deriving underestimator and overestimator functions that can be incorporated into an optimization algorithm.

Let z in equation 2.14 be a product of two continuous variables, x and y.

$$z = xy \qquad x, y \in R \tag{2.14}$$

Each continuous variable has a lower and an upper bound as illustrated by constraint 2.15 and 2.16.

$$x^L \le x \le x^U \tag{2.15}$$

$$y^L \le y \le y^U \tag{2.16}$$

Constraints 2.17 to 2.20 can be obtained from constraint 2.15 and 2.16.

$$x - x^L \ge 0 \tag{2.17}$$

$$x^U - x \ge 0 \tag{2.18}$$

$$y - y^L \ge 0 \tag{2.19}$$

$$y^U - y \ge 0 \tag{2.20}$$

Constraint 2.21 to 24 are obtained by taking a product of different combinations of constraints 2.17 to 2.20.

$$xy - x^L y - y^L x + x^L y^L \ge 0 \tag{2.21}$$

$$xy - x^U y - y^U x + x^U y^U \ge 0 \tag{2.22}$$

$$xy - x^L y - y^U x + x^L y^U \ge 0 \tag{2.23}$$

$$xy - x^U y - y^L x + x^U y^L \ge 0 \tag{2.24}$$

By substituting equation 2.14 and rearranging, we get constraints 2.25 to 2.28 which are Mccormick (1976) overestimators and underestimators. This method replaces bilinear terms with linear constraints. It is however not an exact linearization technique.

$$z \geq x^L y + y^L x - x^L y^L \qquad 2.25$$

$$z \geq x^U y + y^U x - x^U y^U \qquad 2.26$$

$$z \geq x^L y + y^U x - x^L y^U \qquad 2.27$$

$$z \geq x^U y + y^L x - x^U y^L \qquad 2.28$$

Product of discrete and continuous variables

Glover (1975) presented a method for linearizing a bilinear term due to a product of a discrete and a continuous variable.

Let Z be a product of a discrete variable y and a continuous variable x, as shown in equation 2.29.

$$Z = xy \qquad x \in R, y \in [0,1] \qquad 2.29$$

Z can, therefore, take the value 0 if y is 0 and take the value of x if y is 1. x is a continuous variable with a lower and an upper bound as illustrated by constraint 2.30.

$$x^L \leq x \leq x^U \qquad 2.30$$

Constraint 2.31 is obtained by multiplying constraint 2.30 with the discrete variable y. Constraint 2.32 is obtained by substituting equation 2.29 in constraint 2.31

$$x^L y \leq xy \leq x^U y \qquad 2.31$$

$$x^L y \leq Z \leq x^U y \qquad 2.32$$

$$x - x^U (1-y) \leq Z \leq x + x^L (1-y) \qquad 2.33$$

The lower and the upper bound are assumed to be known, therefore constraints 2.32 and 2.33 are linear in terms of x and y. Equation 2.29 can be linearized by replacing it with constraint 2.32 and 2.33. This method is an exact transformation technique.

Product of discrete variables

$$z = y_1 y_2 \qquad y_1, y_2 \in [0,1] \qquad 2.34$$

A binary variable is an integer (discrete) variable that can only assume a value of zero or one. In optimization formulations, binary variables can be used to model the presence or absence of tasks. Table 2.2 shows possible outcomes of equation 2.34, a product of two binary variables. This outcome shows the activity/inactivity of a task that requires both y_1 and y_2 to be present.

Table 2.2 Product of two binary variables

y_1	y_2	z
1	1	1
1	0	0
0	1	0
0	0	0

Nonlinearity exists in optimization formulations can be due to the product of binary variables. If z is a product of two binary variables, the following set of linear constraints can replace equation 2.34 (Maranas and Zomorrodi, 2016):

$$z \le y_1 \qquad 2.35$$

$$z \le y_2 \qquad 2.36$$

$$z \ge y_1 + y_2 - 1 \qquad 2.37$$

Equation 2.35 and 2.36 provide the upper bound for z and also hold for all combinations in Table 2.2. Equation 2.37 provides the lower bound for z. This exact linearization technique can be expanded for the product of any number of binary variables. Z is a product of any number of binary variables, equation 2.38.

$$Z = \prod_{i=1}^{N} y_i \tag{2.38}$$

The following set of general linear constraints can replace 2.38 (Maranas and Zomorrodi, 2016):

$$Z \geq 0 \tag{2.39}$$

$$Z \leq y_i \qquad \forall i \in \{1,2,\ldots,N\} \tag{2.40}$$

$$Z \geq \sum_{i=1}^{N} y_i - (N-1) \tag{2.41}$$

2.7.5. Solution output

It is usual to obtain an unacceptable solution output when running a mathematical model. An unacceptable solution output can include solver failure, infeasible solution, unbounded solution and unsatisfactory optimal solution. Solver failure can occur when a solver fails to cite numerical difficulties; when the unrealistically large amount of resources (memory and time) are used to make little progress; and cycling where a model lacks progress as it iterates excessively at a single point despite using more resources. A solver can sometimes stop and indicate that the model is infeasible or unbounded when attempting a model solution. Sometimes an optimal solution can be reported while the values of variables are observed to be impractical. This unsatisfactory optimal solution may be because of omitted variables or constraints, errors in estimated parameters, algebraic errors, etc.

Solver failure can be alleviated by examining the model structure and input coefficient location, by using a priori degeneracy resolution scheme (adding small numbers to one side of the equation to avoid redundancy) and/or by rescaling the model to narrow the disparity between the magnitude of variable coefficients (Mccarl and Spreen, 2011). These techniques should be applied before solving the model to avoid solver failure. Unbounded solutions can be alleviated by imposing upper bounds to variables that are taking undesirable outcomes. For infeasible solutions, structural checking can be done to find obvious formulation defects or by using artificial variables that make infeasible problems feasible by allowing the violation of equality constraints. This then makes it easier to discover constraints casing infeasibility.

2.8. Remarks

Rapid-changing markets have led to an increase in the use of batch manufacturing processes. High water consumption and the degradation of water sources by manufacturing industries contribute significantly to the water scarcity problem. This has triggered the use of process integration techniques, such as direct and indirect water reuse and recycle, to optimize the use of water in batch manufacturing processes. Mathematical models, presented in literature, that use process integration techniques to minimize wastewater in batch processes do not account for sequence dependent changeovers. As a result, they determine the amount of water required for washing operations by only looking at the task that has just taken place in a unit. Incorporating sequence dependent changeover constraints can open an opportunity to explore sequence dependent water saving opportunities. Presented in this work are wastewater minimization formulations for multipurpose batch processes which explore sequence dependent changeover opportunities for water minimization simultaneously with direct and indirect water reuse and recycle opportunities.

2.9. References

Adekola, O., Majozi, T., 2017. Wastewater minimization in batch plants with sequence dependent changeover. Comput. Chem. Eng. 97, 85–103. doi:10.1016/j.compchemeng.2016.11.016

Adekola, O., Majozi, T., 2011. Wastewater minimization in multipurpose batch plants with a regeneration unit: Multiple contaminants. Comput. Chem. Eng. 35, 2824–2836. doi:10.1016/j.compchemeng.2011.04.008

Adekola, O., Stamp, J.D., Majozi, T., Garg, A., 2013. Unified Approach for the Optimization of Energy and Water in Multipurpose Batch Plants Using a Flexible Scheduling Framework.

Almató, M., Sanmartí, E., Espun˜a, A., Puigjaner, L., 1997. Rationalizing the water use in the batch process industry. Comput. Chem. Eng. 21, S971–S976. doi:10.1016/S0098-1354(97)87628-4

Buabeng-Baidoo, Mafukidze N., Pal J., Tiwari S., Srinivasan B., Majozi T., Srinivasan R., 2017, Study of water reuse opportunities in a large-scale milk processing plant through process integration, Chemical Engineering Research and Design, 121, pp 81–91, 2017.

Cerda, J., Henning, G.P., Grossmann, I.E., 1997. Mixed-integer linear programming model for short-term scheduling of single-stage multiproduct batch plants with parallel lines. Ind. Eng. Chem. Res. 36, 1695. doi:10.1021/ie9605490

Chaturvedi, N.D., Bandyopadhyay, S., 2014. Simultaneously targeting for the minimum water requirement and the maximum production in a batch process. J. Clean. Prod. 77, 105–115. doi:10.1016/j.jclepro.2013.11.079

Chen, C.-L., Chang, C.-Y., 2009. A resource-task network approach for optimal short-term/periodic scheduling and heat integration in multipurpose batch plants. Appl. Therm. Eng. 29, 1195–1208. doi:10.1016/j.applthermaleng.2008.06.014

Chen, C.-L., Chang, C.-Y., Lee, J.-Y., 2011. Resource-Task Network Approach to Simultaneous Scheduling and Water Minimization of Batch Plants. Ind. Eng. Chem. Res. 50, 3660–3674. doi:10.1021/ie1007536

Chen, C.L., Chang, C.Y., Lee, J.Y., 2008. Continuous-time formulation for the synthesis of water-using networks in batch plants. Ind. Eng. Chem. Res. 47, 7818–7832. doi:10.1021/ie800573r

Cheng, K.-F., Chang, C.-T., 2007. Integrated Water Network Designs for Batch Processes. Ind. Eng. Chem. Res. 46, 1241–1253. doi:10.1021/ie0611150

Chew, I.M.L., Tan, R., Ng, D.K.S., Foo, D.C.Y., Majozi, T., Gouws, J., 2008. Synthesis of direct and indirect interplant water network. Ind. Eng. Chem. Res. 47, 9485–9496. doi:10.1021/ie800072r

Chwan, D., Foo, Y., 2009. State-of-the-Art Review of Pinch Analysis Techniques for Water Network Synthesis. Ind. Eng. Chem. Res 5125–5159. doi:10.1021/ie801264c

Cobelli, C., Carson, E., 2001. Chapter 1 – An Introduction to Modelling Methodology, in: Modeling Methodology for Physiology and Medicine. pp. 1–13. doi:10.1016/B978-012160245-1/50002-7

Doganis, P., Sarimveis, H., 2007. Optimal scheduling in a yogurt production line based on mixed integer linear programming. J. Food Eng. 80, 445–453. doi:10.1016/j.jfoodeng.2006.04.062

Duran, M., Grossmann, I., 1986. An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. Math. Program. 36, 307–339. doi:307-339

Dym, C.L., 2004. What Is Mathematical Modeling? 3–12. doi:10.1016/B978-012226551-8/50002-8

Edgar, T.F., Himmelblau, D.M., 1989. Optimization of Chemical Processes. McGraw-Hill.

El-Halwagi, M.M., 2012. Sustainable Design Through Process Integration: Fundamentals and Applications to Industrial Polution Prevention, Resource Conservation, and Profitability Enhancement. Waltham, MA: Elsevier Inc, Texas AandM University.

Erdirik-Dogan, M., Grossmann, I.E., 2008. Slot-based formulation for the short-term scheduling of multistage, multiproduct batch plants with sequence-dependent changeovers. Ind. Eng. Chem. Res. 47, 1159–1163. doi:10.1021/ie070941l

Floudas, C. a., Lin, X., 2004. Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review. Comput. Chem. Eng. 28, 2109–2129. doi:10.1016/j.compchemeng.2004.05.002

Glover, F., 1975. Formulations of Nonlinear Integer. Manage. Sci. 22.

Gouws, J., Majozi, T., Foo, D.C.Y., Chen, C.L., Lee, J.Y., 2010. Water minimization techniques for batch processes. Ind. Eng. Chem. Res. 49, 8877–8893. doi:10.1021/ie100130a

Grundemann, L., Gonschorowski, V., Fischer, N., Scholl, S., 2012. Cleaning waste minimization for multiproduct plants: Transferring macro batch to micro conti manufacturing. J. Clean. Prod. 24, 92–101. doi:10.1016/j.jclepro.2011.11.010

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

Harjunkoski, I., Maravelias, C.T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassick, J., 2014. Scope for industrial applications of production scheduling models and solution methods. Comput. Chem. Eng. 62, 161–193. doi:10.1016/j.compchemeng.2013.12.001

Huang, C.-H., Chang, C.-T., Ling, H.-C., Chang, C.-C., 1999. A Mathematical Programming Model for Water Usage and Treatment Network Design. Ind. Eng. Chem. Res. 38, 2666–2679. doi:10.1021/ie990043s

Ierapetritou, M.G., Floudas, C. a., 1998. Effective Continuous-Time Formulation for Short-Term Scheduling. 1. Multipurpose Batch Processes. Ind. Eng. Chem. Res. 37, 4341–4359. doi:10.1021/ie970927g

Janak, S.L., Floudas, C.A., 2008. Improving unit-specific event based continuous-time approaches for batch processes: Integrality gap and task splitting. Comput. Chem. Eng. 32, 913–955. doi:10.1016/j.compchemeng.2007.03.019

Janak, S.L., Lin, X., Floudas, C. a, 2004. Enhanced Continuous-Time Unit-Specific Event-Based Formulation for Short-Term Scheduling of Multipurpose Batch Processes: Resource Constraints and Mixed Storage Policies. Ind. Eng. Chem. Res. 43, 2516.

Kabra, S., Shaik, M.A., Rathore, A.S., 2013. Multi-period scheduling of a multi-stage multi-product bio-pharmaceutical process. Comput. Chem. Eng. 57, 95–103. doi:10.1016/j.compchemeng.2013.03.009

Kim, J.-K., Smith, R., 2004. Automated Design of Discontinuous Water Systems. Process Saf. Environ. Prot. 82, 238–248. doi:10.1205/095758204323066000

Kondili, E., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations—II. Computational issues. Comput. Chem. Eng. 17, 229–244. doi:10.1016/0098-1354(93)80016-G

Lee, S.J., Wan Alwi, S.R., Lim, J.S., Manan, Z.A., 2014. Minimum water network design for fixed schedule and cyclic operation batch processes with minimum storage capacity and inter-connections. J. Clean. Prod. 77, 65–78. doi:10.1016/j.jclepro.2013.11.061

Li, B.H., Chang, C.T., 2006. A mathematical programming model for discontinuous water-reuse system design. Ind. Eng. Chem. Res. 45, 5027–5036. doi:10.1021/ie050421v

Li, J., Floudas, C. a., 2010. Optimal Event Point Determination for Short-Term Scheduling of Multipurpose Batch Plants via Unit-Specific Event-Based Continuous-Time Approaches. Ind. Eng. Chem. Res. 49, 7446–7469. doi:10.1021/ie901842k

Li, J., Susarla, N., Karimi, I. a., Shaik, M. a., Floudas, C. a., 2010. An analysis of some unit-specific event-based models for the short-term scheduling of noncontinuous processes. Ind. Eng. Chem. Res. 49, 633–647. doi:10.1021/ie801879n

Linnhoff, B., Hindmarsh, E., 1983. The pinch design method for heat exchanger networks. Chem. Eng. Sci. 38, 745–763. doi:10.1016/0009-2509(83)80185-7

Lundell, A., Westerlund, T., 2012. Global optimization of mixed-integer signomial programming problems. IMA Vol. Math. its Appl. 154. doi:10.1007/978-1-4614-1927-3_12

Majozi, T., 2010. Batch Chemical Process Integration. doi:10.1007/978-90-481-2588-3

Majozi, T., 2005a. Wastewater minimisation using central reusable water storage in batch plants. Comput. Chem. Eng. 29, 1631–1646. doi:10.1016/j.compchemeng.2005.01.003

Majozi, T., 2005b. An effective technique for wastewater minimisation in batch processes. J. Clean. Prod. 13, 1374–1380. doi:10.1016/j.jclepro.2005.04.016

Majozi, T., Brouckaert, C.J., Buckley, C. a., 2006. A graphical technique for wastewater minimisation in batch processes. J. Environ. Manage. 78, 317–329. doi:10.1016/j.jenvman.2005.04.026

Majozi, T., Gouws, J.F., 2009. A mathematical optimisation approach for wastewater minimisation in multipurpose batch plants: Multiple contaminants. Comput. Chem. Eng. 33, 1826–1840. doi:10.1016/j.compchemeng.2009.06.008

Majozi, T., Seid, E.R., Lee, J.-Y., 2015. Synthesis, Design, and Resource Optimization in Batch Chemical Plants. Taylor & Francis Group, LLC.

Majozi, T., Zhu, X.X., 2001. Novel continuous-time MILP formulation for multipurpose batch plants. 1. Short-term scheduling. Ind. Eng. Chem. Res. 40,

5935. doi:10.1021/ie0005452

Maranas, C.D., Zomorrodi, A.R., 2016. Optimization Methods in Metabolic Networks. Wiley.

Maravelias, C.T., Grossmann, I.E., 2003. New General Continuous-Time State - Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants 3056–3074.

Mccarl, B.A., Spreen, T.H., 2011. Applied Mathematical Programming Using Algebraic Systems. Texas A&M Univ. 4, 425–438. doi:http://agecon2.tamu.edu/people/faculty/mccarl-bruce/books.htm

Mccormick, G.P., 1976. Computability of global solutions to factorable nonconvex programs: part i -convex underestimating problems *. Math. Program. 10, 147–175.

Méndez, C.A., Henning, G.P., Cerdá, J., 2001. An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. Comput. Chem. Eng. 25, 701–711. doi:10.1016/S0098-1354(01)00671-8

Méndez, C. a., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput. Chem. Eng. 30, 913–946. doi:10.1016/j.compchemeng.2006.02.008

Nonyane, D.R., Majozi, T., 2012. Long term scheduling technique for wastewater minimisation in multipurpose batch processes. Appl. Math. Model. 36, 2142–2168. doi:10.1016/j.apm.2011.08.007

Oliver, P., Rodríguez, R., Udaquiola, S., 2008. Water use optimization in batch process industries. Part 1: design of the water network. J. Clean. Prod. 16, 1275–1286. doi:10.1016/j.jclepro.2007.06.012

Pantelides, C., 1994. Unified frameworks for optimal process planning and scheduling. Proc. Second ….

Robin Smith, 2014. Wiley: CChemical Process: Design and Integrationhemical Process: Design and Integration - Robin Smith.

Ryoo, H., Sahinidis, V., 1996. A Branch-and-Reduce Approach to Global Optimization. J. Glob. Optim. 107–138. doi:107-138

Sahinidis, N. V, 2000. Branch And Reduce Optimization Navigator. Univ. Illinois Urbana-Champaign 217–244.

Schilling, G., Pantelides, C.C., 1996. A simple continuous-time process scheduling formulation and a novel solution algorithm. Comput. Chem. Eng. 20, S1221–S1226. doi:10.1016/0098-1354(96)00211-6

Seid, E.R., Majozi, T., 2014. Optimization of energy and water use in multipurpose batch plants using an improved mathematical formulation. Chem. Eng. Sci. 111, 335–349. doi:10.1016/j.ces.2014.02.036

Seid, R., Majozi, T., 2012. A robust mathematical formulation for multipurpose batch plants. Chem. Eng. Sci. 68, 36–53. doi:10.1016/j.ces.2011.08.050

Shah, N., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations—II. Computational issues. Comput. Chem. Eng. 17, 229–244. doi:10.1016/0098-1354(93)80016-G

Shaik, M.A., Floudas, C.A., Kallrath, J., Pitz, H.J., 2009. Production scheduling of a large-scale industrial continuous plant: Short-term and medium-term scheduling. Comput. Chem. Eng. 33, 670–686. doi:10.1016/j.compchemeng.2008.08.013

Shaik, M.A., Vooradi, R., 2013. Uni fi cation of STN and RTN based models for short-term scheduling of batch plants with shared resources. Chem. Eng. Sci. 98, 104–124. doi:10.1016/j.ces.2013.05.019

Shaik, M. a., Floudas, C. a., 2009. Novel unified modeling approach for short-term scheduling. Ind. Eng. Chem. Res. 48, 2947–2964. doi:10.1021/ie8010726

Shaik, M. a., Floudas, C. a., 2008. Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using RTN framework. Comput. Chem. Eng. 32, 260–274. doi:10.1016/j.compchemeng.2007.05.007

Shaik, M. a., Janak, S.L., Floudas, C. a., 2006. Continuous-time models for short-term scheduling of multipurpose batch plants: A comparative study. Ind. Eng. Chem. Res. 45, 6190–6209. doi:10.1021/ie0601403

Sherali, H.D., Adams, W.P., 1994. A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. Discret. Appl. Math. 52, 83–106. doi:10.1016/0166-218X(92)00190-W

Smith, E.., 1996. On the optimal design of continuous processes. Ph.D. Dissertation, under supervision of C. Pantelides. ImperialCollege of Science, Technology and Medicine, London, UK.

Sundaramoorthy, A., Karimi, I. a., 2005. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. Chem. Eng. Sci. 60, 2679–2702. doi:10.1016/j.ces.2004.12.023

Tchobanoglous, G., Abu-Orf, M., Burton, F.L., Bowden, G., Eddy, M., Stensel, H.D., Pfrang, W., Education, M.-H., 2014. Wastewater Engineering: Treatment and Resource Recovery, Volume 2. McGraw-Hill Education.

Vooradi, R., Shaik, M. a., 2013. Rigorous unit-specific event-based model for short-term scheduling of batch plants using conditional sequencing and unit-wait times. Ind. Eng. Chem. Res. 52, 12950–12972. doi:10.1021/ie303294k

Wang, Y., Smith, R., 1995. Time Pinch Analysis. Chem. Eng. Res. Des. 73, 905–914.

Wang, Y.P., Smith, R., 1994. Wastewater minimisation. Chem. Eng. Sci. 49, 981–1006. doi:10.1016/0009-2509(94)80006-5

Williams, H., 1999. Model Building in Mathematical Programming, 4th Edition.

Yeomans, H., Grossmann, I.E., 1999. A systematic modeling framework of superstructure optimization in process synthesis 23, 709–731.

Chapter 3

MODEL DEVELOPMENT

3.1. Introduction

This chapter presents the development of the optimization mathematical formulations for water minimization in multipurpose batch plants. Four different scenarios are considered: fixed water requirement with sequence dependent changeover constraints, fixed outlet concentration with sequence dependent changeover constraints, fixed water requirement with sequence dependent changeover constraints and water reuse and recycle technique, as well as fixed outlet concentration with sequence dependent changeover constraints and water reuse and recycle technique. This is followed by designed superstructures, which are based on the problem statement presented in Chapter 1. Assumptions made when developing the model are presented as well as the nomenclature. Lastly, mathematical formulations are presented for the scenarios under consideration together with the objective function that maximizes the profitability of the process across the time horizon of interest.

3.2. Explored scenarios

In order to incorporate sequence dependent changeover constraints in a mathematical formulation, a sequence dependent parameter is required. In this work, four scenarios based on sequence dependent parameter and the superstructures in Figures 3.1 and 3.2 were considered. The first scenario is based on fixed sequence-dependent changeover water requirement of each washing operation while the outlet concentration was allowed to vary. This scenario can be applied to both single and multiple contaminant problems. In the second scenario, the outlet concentration is fixed and the washing water requirement determined. The sequence-dependent parameter is a fraction or percentage used to determine the additional amount of water that must be used to rinse the processing unit depending the sequence of tasks in the unit. The fixed outlet concentration scenario cannot be extended to multiple contaminant problems. This is because the outlet concentrations of individual components cannot all be set to a maximum, since contaminants cannot be limiting simultaneously.

The third and the fourth scenarios respectively expand on the first and second scenarios by exploring sequence-dependent changeover opportunities for water minimization simultaneously with water reuse and recycle. One of the major challenges in mathematical optimization is obtaining the accurate data to feed into the model in order to obtain reliable predictions. Developing formulations that explore the same concept with a similar objective but require different data increases the chances of benefiting from that concept.

3.3. Superstructure representation

Figure 3.1 represents a superstructure for sequence-dependent changeover where different tasks can be processed in a multipurpose unit j. The amount of water required for a washing operation differs with the sequence of tasks. For instance, $w_z(s_{inj}, s_{inj})$ amount of water is required when task s_{inj} follows task s_{inj},

$w_z(s_{inj}, s''_{inj})$ amount of water is required when task s''_{inj} follows task s_{inj}, and $w_z(s''_{inj}, s'_{inj})$ amount is required when task s''_{inj} follows task s'_{inj}. Within a given time horizon of interest, the model must synthesize a sequence of tasks that will generate the least amount of wastewater. A trade-off, therefore, exists between production and wastewater minimization.

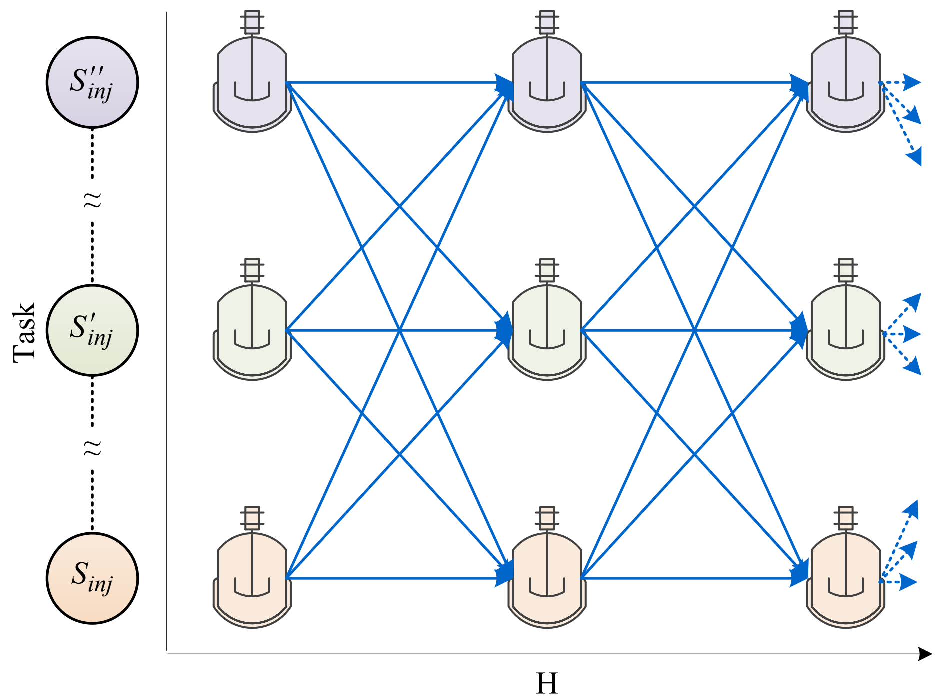


Figure 3.1 Superstructure for sequence dependent changeover opportunity for water minimization

Figure 3.2 is a superstructure for a water minimization problem with a central water storage illustrating both direct and indirect water reuse and recycle opportunities. Water required for washing operation j is not only freshwater $mw_f(s_{inj}, p)$ but could also be indirectly reused or recycled from the central reusable storage tank as $mw_s^{out}(s_{inj}, p)$ and/or directly reused from other washing operations in other processing units j'. Similarly, the outlet stream from a washing operation can be disposed of as effluent $mw_e(s_{inj}, p)$, can be sent to the reusable water tank as

$mw_s^{in}(s_{inj}, p)$ and/or directly reused to another washing operation in other process units *j'* as $mw_r(s_{inj}, s'_{inj'}, p)$.

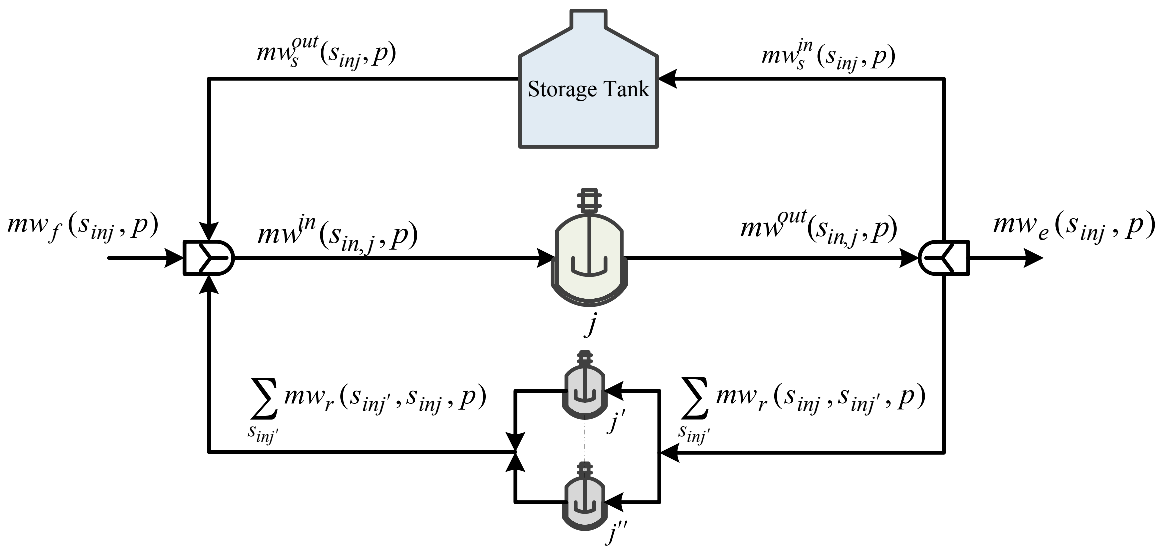


Figure 3.2 Superstructure for direct and indirect water reuse and recycle

3.4. Assumptions

The following assumptions were made when developing the proposed mathematical model:

- The entire mass load in a unit is removed when the washing operation is complete.
- The mass load does not become difficult to wash if left in a unit for a period of time.
- Washing operations are required after processing the last task in a unit.
- The inlet and outlet contaminant concentrations parameters are the maximum allowable.
- Freshwater has no contaminants.
- Sequence independent changeovers are lumped in the processing time of tasks.

- The changeover time is the duration of changeover washing operations.

3.5. Nomenclature

A list of sets, parameters and variables used by the developed mathematical formulations are presented below:

Sets

S	$\{s \mid s$ Any state$\}$
P	$\{p \mid p$ Time point$\}$
J	$\{j \mid j$ Processing unit$\}$
S_{inj}	$\{s_{inj} \mid s_{inj}$ Effective state representing a task performed in a unit$\}$
S_p	$\{s_p \mid s_p$ Product$\}$
K	$\{k \mid k$ Contaminants$\}$

Parameters

$V_{s_{inj}}^{L}$	Lower bound in capacity of a given unit that processes the effective state s_{inj}
$V_{s_{inj}}^{U}$	Upper bound in capacity of a given unit that processes the effective state s_{inj}
V_{j}^{U}	Maximum capacity of unit j
$\rho_{s_{inj}}^{sc}$	Portion of state s consumed by a task that processes the effective state s_{inj}
$\rho_{s_{inj}}^{sp}$	Portion of state s produced by a task that processes the effective state s_{inj}
$\alpha(s_{inj})$	Constant coefficient of processing time of task that processes the effective state s_{inj}
$\beta(s_{inj})$	Variable coefficient of processing time of task that processes the effective state s_{inj}
$W_z(s_{inj}, s'_{inj})$	Amount of water required to wash unit j when task $s'_{in,j}$ follows task s_{inj}

$W_h(s_{inj})$	Amount of water required to wash unit j when s_{inj} is the last task to be processed in that unit.
$WR(s_{inj}, s'_{inj})$	A fraction of water required to clean unit j that will be added for rinsing when task s'_{inj} follows task s_{inj}
$L(s_{inj}, k)$	Contaminant concentration of state that will be left in a unit after processing a task s_{inj}
$W^U(s_{in,j})$	Upper bound of the amount of water for cleaning unit j
$C_w^U(s_{inj}, k)$	Upper bound of the allowable contaminant concentration
$C_s^{in}(k)$	Initial contaminant concentration of water in the storage
Q_s^{in}	Initial amount of water in the storage tank
Q_s^U	Maximum storage capacity
$SP(s_p)$	Selling price of state s
$W_f^{\cos t}$	Cost of freshwater
$W_e^{\cos t}$	Cost of wastewater
D_w	Density of water
R_t	Volumetric flowrate of cleaning sprays
H	Time horizon of interest

Binary variables

$z(s_{inj}, s'_{inj}, p+1, p)$	Binary variable for the changeover from s_{inj} at p to s'_{inj} at $p+1$
$h(s_{inj}, p)$	Binary variable indicating that $s_{in,j}$ is the last task to occur in unit j at p.

$y(s_{inj}, p)$	Binary variable associated with the usage of state s in unit j at p.
$y_r(s_{inj}, s_{inj'}, p)$	Binary variable for the transfer of water from unit j to unit j' at p.
$y_s^{in}(s_{inj}, p)$	Binary variable for the transfer of water from unit j to storage at p.
$y_s^{out}(s_{inj}, p)$	Binary variable for the transfer of water to unit j from storage at p.

Continuous Variables

$mw^{in}(s_{inj}, p)$	Mass of water into unit j at time point p
$mw^{out}(s_{inj}, p)$	Mass of water from unit j at time point p
$mw_f(s_{inj}, p)$	Mass of freshwater into unit j at time point p
$mw_a^{in}(s_{inj}, p)$	Mass of water into unit j at time point p at stage A of the washing operation
$mw_a^{out}(s_{inj}, p)$	Mass of water into unit j at time point p at stage A of the washing operation
$mw_a^f(s_{inj}, p)$	Mass of freshwater into unit j at time point p at stage A of the washing operation
$mw_e(s_{inj}, p)$	Mass of effluent from unit j at time point p
$mw_s^{in}(s_{inj}, p)$	Mass of water transferred to storage from unit j at time point p
$mw_s^{out}(s_{inj}, p)$	Mass of water transferred from storage to unit j at time point p
$mw_r(s_{inj}, s'_{inj'}, p)$	Mass of water transferred from unit j to unit j' at time point p
$cw_a^{in}(s_{inj}, p)$	Inlet concentration to stage A of the washing operation
$cw_a^{out}(s_{inj}, p)$	Outlet concentration to stage A of the washing operation

$cw^{in}(s_{inj}, k, p)$	Contaminant concentration of water into unit j at p
$cw^{out}(s_{inj}, k, p)$	Contaminant concentration of water from unit j at p
$c_s(p, k)$	Contaminant concentration of water in the storage tank at time point p
$q_s(s_{inj}, p)$	Amount of water in the storage tank at time point p
$m(s_{inj}, k, p)$	Contaminant load to be removed by a washing operation in unit j at p
$sr(s, p)$	Amount of state stored at time point p
$mu(s_{inj}, p)$	Total mass of material processed in unit j at time point p
$tw_d(s_{inj}, p)$	The duration of a washing operation in unit j at time point p
$tw^{in}(s_{inj}, p)$	The starting time of a washing operation in unit j at time point p
$tw^{out}(s_{inj}, p)$	The finishing time of a washing operation in unit j at time point p
$tw_s^{in}(s_{inj}, p)$	The time at which water is transferred to storage from unit j at time point p
$tw_s^{out}(s_{inj}, p)$	The time at which water is transferred from storage to unit j at time point p
$tu(s_{inj}, p)$	The starting time of a process task in unit j at time point p
$tp(s_{inj}, p)$	The finishing time of a process task in unit j at time point p
$vw(s_{inj}, p)$	Volume of water into unit j at time point p
$xw(s_{inj}, p)$	Combined mass of water for cleaning and rinsing unit j at time point p
$xc(s_{inj}, p)$	Contaminant concentration of $xw(s_{inj}, p)$

3.6. Mathematical model

Figure 3.3 illustrates the different sub-models that make up the overall structure of the presented models. Mathematical formulations of considered scenarios have different combinations of the following sub-models: scheduling sub-model, sub-model for sequence dependent water saving opportunity, and the sub-model for water reuse and recycle technique with a central water storage tank. Formulations for scenarios 1 and 2 have the scheduling and the sub-model for sequence dependent water saving opportunity. Formulations for scenarios 3 and 4 combine all three sub-models.

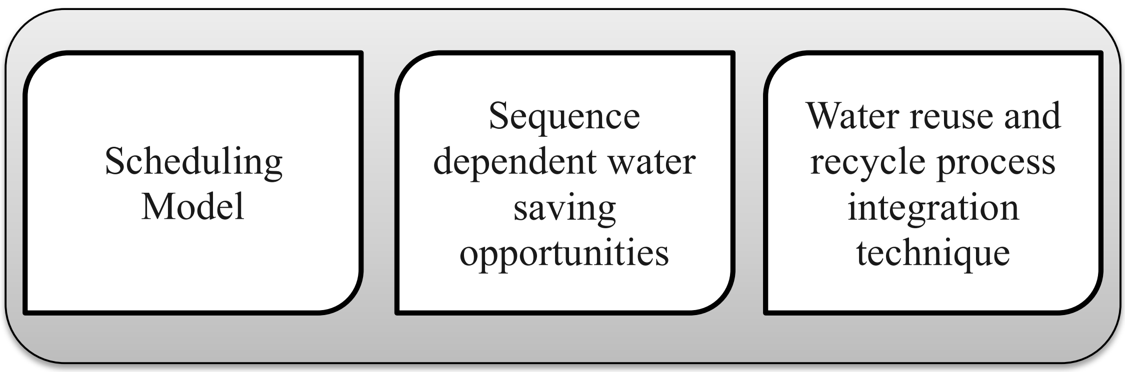


Figure 3.3 Elements of the proposed formulations

3.6.1. Scheduling

True optimality in batch process formulations can only be realized if the production schedule is allowed to vary (Gouws et al., 2008). A scheduling model by Seid and Majozi (2012) was used as a platform when developing variable schedule formulations for all four scenarios. This was because it gave better objective values in a less computational time when compared to other scheduling models in literature. The model of Seid and Majozi (2012) used unit-specific time slots and continuous-time representation and it is based on a State Sequence Network (SSN) that makes use of effective states since they render an opportunity to reduce the number of binary variables (Majozi and Zhu, 2001). Each time slot in the developed model, therefore, represents an unknown duration in which a process task and a washing operation occur.

Allocation constraint

Constraint 3.1 allows only one task to be active in a processing unit j at a given time point p.

$$\sum_{s_{inj}} y(s_{inj}, p) \leq 1, \forall\ p \in P,\ s_{inj} \in S_{inj} \qquad 3.1$$

Capacity constraint

Constraint 3.2 ensures that the amount of batch processed in a unit is within the lower and the upper bounds, $V_{s_{inj}}^{L}$ and $V_{s_{inj}}^{U}$ respectively.

$$V_{s_{inj}}^{L} y(s_{inj}, p) \leq mu(s_{inj}, p) \leq V_{s_{inj}}^{U} y(s_{inj}, p), \forall\ p \in P,\ s_{inj} \in S_{inj} \qquad 3.2$$

Material balance for storage

Constraint 3.3 calculates the amount of material, excluding products, in storage at a given time point as the amount that was there at a previous time point adjusted by the difference between the amount used at a current time point and the amount produced at the previous time point. Constraint 3.4 calculates the amount of product in storage at a time point as a sum of what was available at the previous time point and what is produced at the current time point.

$$q_s(s, p) = q_s(s, p-1) - \sum_{s_{inj}} \rho_{s_{inj}}^{sc} mu(s_{inj}, p) + \sum_{s_{inj}} \rho_{s_{inj}}^{sp} mu(s_{inj}, p-1)$$
$$\forall\ s \in S,\ p \in P, s_{inj} \in S_{inj} \qquad 3.3$$

$$q_s(s_p, p) = q_s(s_p, p-1) + \sum_{s_{inj}} \rho_{s_{inj}}^{sp} mu(s_{inj}, p), \forall p \in P, s^p \in S^p, s_{inj} \in S_{inj} \quad 3.4$$

Duration constraint: Duration as a function of batch size

Equation 3.5 describes the duration of a task, consisting of a fixed and a variable term, added to the starting time of a task when calculating the finishing time of a task.

$$t_p(s_{inj}, p) \geq t_u(s_{inj}, p) + \alpha(s_{inj}) y(s_{inj}, p) + \beta(s_{inj}) mu(s_{inj}, p)$$
$$\forall\, p \in P,\ s_{inj} \in S_{inj} \quad 3.5$$

Sequence constraints

Sequencing same task in the same unit

Constraint 3.6 ensures that a task starts in a unit after the previous task is completed. This constraint applies to similar tasks in a unit.

$$t_u(s_{inj}, p) \geq t_p(s_{inj}, p-1), \forall\ p \in P,\ s_{inj} \in S_{inj} \quad 3.6$$

Sequencing different tasks in the same unit

Constraint 3.7 also ensures that a task starts in a unit after the previous task is completed. This constraint applies to different tasks that are processed in a unit.

$$t_u(s_{inj}, p) \geq t_p(s'_{inj}, p-1), \forall\ p \in P, s_{inj} \neq s'_{inj}, s_{inj}, s'_{inj} \in S_{inj} \quad 3.7$$

Sequencing different tasks in different unit if an intermediate state is produced from one unit

Constraint 3.8 ensures that an intermediate state produced in a unit should not exceed the allowed storage if it is not consumed in another unit ($t(j,p)=0$). However, if the intermediate state is consumed in another unit ($t(j,p)=1$), then the amount stored is less than the amount produced. Constraint 3.9 ensures that the starting time of the consuming task is greater than the finishing time of the task producing the intermediate state.

$$\rho_{s_{inj}}^{sp} mu(s_{in,j}, p-1) \leq q_s(s,p) + V_j^U t(j,p), \forall\, j \in J,\, p \in P,\, s_{inj} \in S_{inj}^{sp} \quad 3.8$$

$$t_u(s_{inj'}, p) \geq t_p(s_{inj}, p-1) - H(2 - y(s_{inj}, p-1) - t(j,p))$$
$$\forall\, j \in J,\, p \in P,\, s_{inj}, s_{inj'} \in S_{inj} \quad 3.9$$

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

Constraint 3.10 allows the state used by a task at time point p to come from other units that produced the same state at a previous time point. Constraint 3.11 ensures that a task consuming a state occurs after the completion of the producing tasks.

$$\sum_{s_{inj}} \rho_{s_{inj}}^{sc} mu(s_{inj}, p) \leq q_s(s, p-1) + \sum_{s_{inj}} \rho_{s_{inj}}^{sp} mu(s_{in,j}, p-1) t(j,p)$$
$$\forall\, j \in J,\, p \in P,\, s_{inj} \in S_{inj} \quad 3.10$$

$$t_u(s_{inj'}, p) \geq t_p(s_{inj}, p-2) - H(1 - y(s_{inj}, p-2))$$
$$\forall p \in P,\, s_{inj}, s_{inj'} \in S_{inj} \quad 3.11$$

Constraints for FIS policy

Constraint 3.12 ensures that the produced state is immediately consumed or not produced at all if there is no storage capacity available. Constraint 3.13 ensures that the finishing time of the producing task coincides with the starting time of the consuming task.

$$\sum_{s_{inj}} \rho_{s_{inj}}^{sp} mu(s_{inj}, p-1) + q_s(s, p-1) \le QS^U + \sum_j V_j^U (1 - x(s,p)) \quad \forall\ j \in J,\ p \in P,\ s \in S, s_{inj} \in S_{inj} \tag{3.12}$$

$$t_u(s_{inj'}, p) \le t_p(s_{inj}, p-1) + H(2 - y(s_{inj'}, p) - y(s_{inj}, p-1)) + H(x(s,p)) \quad \forall p \in P,\ s_{inj}, s_{inj'} \in S_{inj},\ s \in S \tag{3.13}$$

Storage constraints when idle unit stores material produced previously

Constraint 3.14 ensures that material produced can be stored in a storage unit with a maximum capacity and/or in a processing unit that produced it if that unit is not processing a task in the next time point. Constraint 3.15 ensures that materials are stored for consecutive time points in a processing unit. Constraint 3.16 prevents the processing unit from starting a task at the time point when materials are stored.

$$q_s(s,p) \le QS^U + \sum_{s_{inj}} u(s_{inj}, p), \forall p \in P,\ s_{inj} \in S_{inj}, s \in S \tag{3.14}$$

$$u(s_{inj}, p) \le \rho_{s_{inj}}^{sp} m_u(s_{inj}, p-1) + u(s_{inj}, p-1), \forall\ p \in P,\ s_{inj} \in S_{inj} \tag{3.15}$$

$$u(s_{inj}, p) \le V_j^U - \left(1 - \sum_{S_{inj}} y(s_{inj}, p)\right), \forall\ j \in J,\ p \in P,\ s_{inj} \in S_{inj} \tag{3.16}$$

Time horizon constraints

Constraint 3.17 and 3.18 ensures that all tasks are processed within the time horizon of interest.

$$t_u(s_{inj}, p) \leq H \ , \forall p \in P, s_{inj} \in S_{inj} \qquad 3.17$$

$$t_p(s_{inj}, p) \leq H \ , \forall p \in P, s_{inj} \in S_{inj} \qquad 3.18$$

3.6.2. Scenario 1: Fixed water requirement with sequence dependent changeover constraints

The formulation for this scenario is based on the superstructure in Figure 3.1.

Changeover constraints

A changeover binary variable $z(s_{inj}, s'_{inj}, p+1, p)$ takes the value of 1 when task s_{inj} occurs at time slot p is followed by task s'_{inj} at $p+1$ in the same unit, as ensured by constraint 3.19. The changeover variable is declared as a continuous variable and can only assume a value of 0 or 1 since it is determined from binary variables as presented in constraint 3.19. Constraint 3.20 ensures that if s_{inj} occurs at a time slot, it can either be followed by a task in the next time slot or it is the last task to occur in that unit. Constraint 3.21 ensures that at any given process unit j, there can only be one last task in a unit.

$$z(s_{inj}, s'_{inj}, p+1, p) = y(s_{inj}, p) y(s'_{inj}, p+1), \forall p \in P, s_{inj}, s'_{inj} \in S_{inj} \qquad 3.19$$

$$\sum_{s'_{inj}} z(s_{inj}, s'_{inj}, p+1, p) + h(s_{inj}, p) = y(s_{inj}, p) \ , \forall p \in P, s_{inj}, s'_{inj} \in S_{inj} \qquad 3.20$$

$$\sum_{P} \sum_{S_{inj}} h(s_{inj}, p) = 1 \ , \forall p \in P, s_{inj} \in S_{inj} \qquad 3.21$$

Figure 3.4(a) illustrates that a changeover exists between two consecutive tasks in a unit and Figure 3.4(b) illustrates that a task is considered the last if no other task is processed after it in a unit.

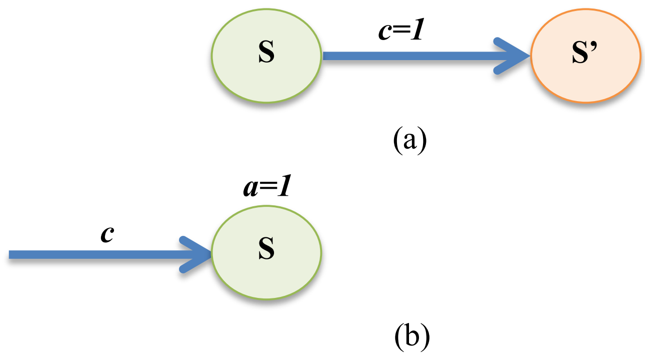


Figure 3.4 (a) First or intermediate task (b) Last task

Constraint 3.19, which consists of a product of two binary variables, is nonlinear and can be linearized by using constraints 3.22, 3.23 and 3.24 (Maranas and Zomorrodi, 2016). Equations 3.22, 3.23 and 3.24 are linear and will ensure that $z(s_{inj}, s'_{inj}, p+1, p)$ takes a value of 1 when task $s_{in,j}$ occurs at time slot p is followed by task s'_{inj} at $p+1$.

$$z(s_{inj}, s'_{inj}, p+1, p) \leq y(s_{inj}, p), \forall p \in P, s_{inj}, s'_{inj} \in S_{inj} \tag{3.22}$$

$$z(s_{inj}, s'_{inj}, p+1, p) \leq y(s'_{inj} p+1), \forall p \in P, s_{inj}, s'_{inj} \in S_{inj} \tag{3.23}$$

$$z(s_{inj}, s'_{inj}, p+1, p) \geq y(s_{inj}, p) + y(s'_{inj}, p+1) - 1 \quad \forall p \in P, s_{inj}, s'_{inj} \in S_{inj} \tag{3.24}$$

Water balance constraints

Constraints 3.25 and 3.26 respectively ensure that all the water used for washing is freshwater and wastewater is disposed of as effluent. Constraint 3.27 is the law of conservation of mass, to ensure that the water that goes into a washing operation equals to water leaving the washing operation. Constraint 3.28 chooses the amount of water to be used for washing operations depending on the sequence of tasks.

$$mw^{in}(s_{inj}, p) = mw_f(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \qquad 3.25$$

$$mw^{out}(s_{in,j}, p) = mw_e(s_{in,j}, p), \forall p \in P, s_{inj} \in S_{inj} \qquad 3.26$$

$$mw^{in}(s_{inj}, p) = mw^o(s_{inj}, p), \ \forall p \in P, s_{inj} \in S_{inj} \qquad 3.27$$

$$mw^{in}(s_{inj}, p) = \sum_{s'_{inj}} W_z(s_{inj}, s'_{inj}) z(s_{inj}, s'_{inj}, p+1, p) + W_h(s_{inj}) h(s_{inj}, p) \qquad 3.28$$
$$\forall p \in P, s_{inj}, s'_{inj} \in S_{inj}$$

Contaminant balance constraints

Constraint 3.29 determines the amount of load to be removed by the washing operation. The binary variables are included when consecutive tasks or the last task does not require a washing operation. For instance, washing may not be required when two consecutive batches of the same task are processed in a unit if the residual material will not contaminate the succeeding batch. Constraint 3.30 ensures that the entire load in a unit is removed by the washing operation and the contaminant concentration of the wastewater is determined. This is because, in this scenario, the fixed water requirement is fixed by constraint 3.28. Constraint 3.31 ensures that the contaminant concentration of the outlet stream does not exceed the maximum allowable. The model reduces the batch size to ensure that the amount of load determined by constraint 3.29 does not result in a contaminant concentration that violates constraint 3.31.

$$m(s_{inj}, k, p) = mu(s_{inj}, p) L(s_{inj}, k) \left(1 - \sum_{s'_{inj}} z(s_{inj}, s'_{inj}, p+1, p) + h(s_{inj}, p) \right. \qquad 3.29$$
$$\forall p \in P, s_{inj}, s'_{inj} \in S_{inj}$$

$$mw^{out}(s_{inj}, p)cw^{out}(s_{inj}, k, p) = m(s_{inj}, k, p), \forall p \in P, s_{inj} \in S_{inj} \qquad 3.30$$

$$cw^{out}(s_{inj}, k, p) \leq C_w^U(s_{inj}, k)y(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \qquad 3.31$$

Sequencing constraints

The duration of washing operations is directly linked to the amount of water required using a fixed volumetric flowrate of the high-pressure water sprays that are used to clean the processing units. Constraint 3.32 determines the duration of washing operations based on the required amount for washing. Constraint 3.33 determines the finishing time of a washing operation by adding the duration of washing to the starting time of the sequence dependent changeover washing operation. Constraint 3.34 states that for a processing task to start in a unit at a time slot, the washing operation that occurred at the last time slot should be complete. Constraint 3.35 states that for a washing operation to occur in a unit at a time slot, the processing task that occurred in the same time slot must be complete. Constraints 3.36 and 3.37 ensure that all washing operations are completed within the time horizon of interest.

$$tw_d(s_{inj}, p)R_t(j) = mw^{out}(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \qquad 3.32$$

$$tw^{out}(s_{inj}, p) = tw^{in}(s_{inj}, p) + tw_d(s_{inj}, p)\left(\sum_{s'_{inj}} z(s_{inj}, s'_{inj}, p+1, p) + h(s_{inj}, p)\right) \qquad 3.33$$
$$\forall p \in P, s_{inj}, s'_{inj} \in S_{inj}$$

$$t_u(s_{inj}, p) \geq tw^{out}(s'_{inj}, p-1), \forall p \in P, s_{inj}, s'_{inj} \in S_{inj} \qquad 3.34$$

$$tw^{in}(s_{inj}, p) \geq t_p(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \qquad 3.35$$

$$tw^{in}(s_{inj}, p) \leq H, \forall p \in P, s_{inj} \in S_{inj} \quad 3.36$$

$$tw^{out}(s_{inj}, p) \leq H, \forall p \in P, s_{inj} \in S_{inj} \quad 3.37$$

3.6.3. Scenario 2: Fixed outlet concentration with sequence dependent changeover constraints

The formulation for this scenario is also based on the superstructure in Figure 3.1. Figure 3.5 illustrates that the washing operations in this scenario explicitly occur in two stages, A and B. Stage A is responsible for removing most of the load and stage B is where rinsing occurs for quality assurance purposes. The outlet contaminant concentration for stage A is fixed to a maximum, and the formulation determines the amount of water required for removing the load such that the maximum concentration is not exceeded. The amount of water required for rinsing in stage B is determined as a fraction or percentage of the amount used in stage A. And because the intensity of rinsing depends on the nature of the products involved in the sequence, the additional rinsing fraction is given as the sequence dependent value in this scenario.

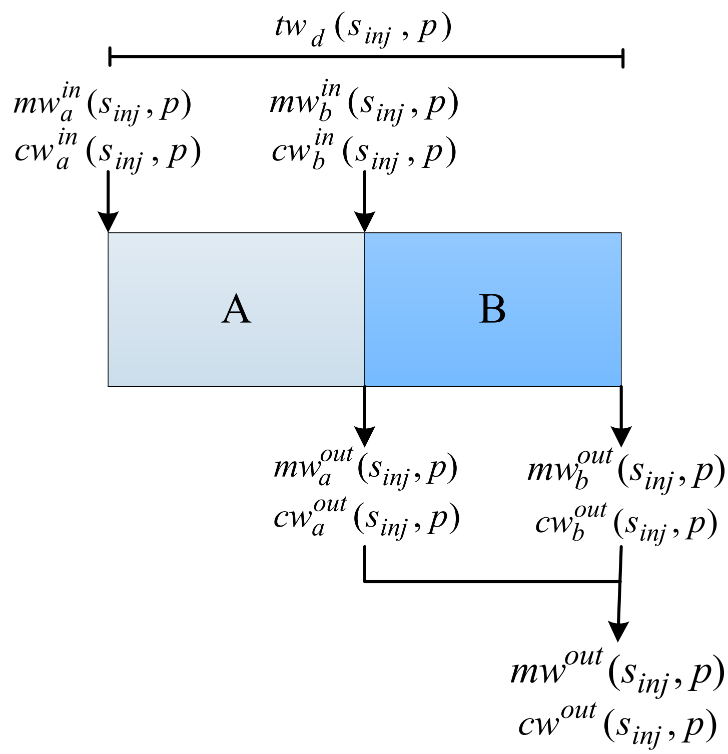


Figure 3.5 Two stages involved in the cleaning operation

Water balance constraints

Constraint 3.38 ensures that the water used in stage A is freshwater. Constraints 3.39 and 3.40 respectively ensure that water in stage A is conserved and does not exceed the maximum allowable. Equation 3.41 determines $mw^{out}(s_{inj}, p)$ which is a combination of the amount used in stage A and the amount of freshwater required for rinsing the processing unit in stage B. Constraint 3.42 ensures that all the water used in both stages is disposed of as effluent.

$$mw_a^{in}(s_{inj}, p) = mw_a^{f}(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \qquad 3.38$$

$$mw_a^{in}(s_{inj}, p) = mw_a^{out}(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \qquad 3.39$$

$$mw_a^{in}(s_{inj}, p) \le W^U(s_{inj})y(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \tag{3.40}$$

$$mw^{out}(s_{inj}, p) = mw_a^{in}(s_{inj}, p) + mw_a^{in}(s_{inj}, p)\sum_{s'_{inj}} AW(s_{inj}, s'_{inj})c(s_{inj}, s'_{inj}, p+1, p) \quad \forall p \in P, s_{inj}, s'_{inj} \in S_{inj} \tag{3.41}$$

$$mw^{out}(s_{inj}, p) = mw_e(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \tag{3.42}$$

Contaminant balance constraints

Constraint 3.29 still hold in this scenario. Constraint 3.43 is the contaminant balance around stage A stating that the load in the tank is removed by the washing water. Constraint 3.44 replaces constraint 3.31 by setting the outlet contaminant concentration to a maximum. Constraint 3.45 simply determines the contaminant concentration $cw^{out}(s_{inj}, p)$.

$$mw_a^{out}(s_{inj}, p)cw_a^{out}(s_{inj}, p) = m(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \tag{3.43}$$

$$cw_a^{out}(s_{inj}, p) = C^U(s_{inj})y(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \tag{3.44}$$

$$mw^{out}(s_{inj}, p)cw^{out}(s_{inj}, p) = mw_a^{out}(s_{inj}, p)cw_a^{out}(s_{inj}, p) \quad \forall p \in P, s_{inj} \in S_{inj} \tag{3.45}$$

3.6.4. Scenario 3: Fixed water requirement with sequence dependent changeover constraints and water reuse and recycle

The formulation for this scenario simultaneously explores sequence dependent changeover opportunity for water minimization with water reuse and recycle opportunities in a multipurpose batch process. The constraints for this scenario are based on both superstructures in Figures 3.1 and 3.2.

Water balance constraints

Constraints 3.27 and 3.28 still hold in this scenario. Constraint 3.46 states that the inlet stream to a washing operation is a combination of freshwater, water from reusable storage and water directly reused from washing operations in other process units. Constraint 3.47 ensures that the outlet stream can be disposed of as effluent, sent to a reusable storage or directly reused to a washing operation in other process units. Constraints 3.48 to 3.50 set the upper bounds for the direct water reuse streams, and streams to and fro the reusable water tank, respectively. Constraint 3.51 ensures that water is not sent to a storage tank at the last time slot.

$$mw^{in}(s_{inj},p) = mw_f(s_{inj},p) + mw_s^{out}(s_{inj},p) + \sum_{s'_{inj}} mw_r(s'_{inj},s_{inj},p) \quad 3.46$$

$$\forall p \in P, s_{inj}, s'_{inj} \in S_{inj}$$

$$mw^{out}(s_{inj},p) = mw_e(s_{inj},p) + mw_s^{in}(s_{inj},p) + \sum_{s'_{inj}} mw_r(s_{inj},s'_{inj},p) \quad 3.47$$

$$\forall p \in P, s_{inj}, s'_{inj} \in S_{inj}$$

$$mw_r(s_{inj},p) \le W^U(s_{inj'})y_r(s_{inj},s_{inj'},p), \forall p \in P, s_{inj} \in S_{inj} \quad 3.48$$

$$mw_s^{in}(s_{inj}, p) \le Q_s^U \, y_s^{in}(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \tag{3.49}$$

$$mw_s^{out}(s_{inj}, p) \le W^U(s_{inj}) y_s^{out}(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \tag{3.50}$$

$$mw_s^{in}(s_{inj}, p) = 0, \forall p = |P|, s_{inj} \in S_{inj} \tag{3.51}$$

Contaminant balance constraints

Constraint 3.29 still holds in this scenario. Constraint 3.52 determines the contaminant mass load of the inlet stream to a washing operation. Equation 3.53 is the contaminant mass balance around a processing unit. It states that the mass of contaminant in the outlet stream is a combination of the contaminant in the inlet stream and the load in the processing unit.

$$mw^{in}(s_{inj}, p) cw^{in}(s_{inj}, k, p) = mw_s^{out}(s_{inj}, p) c_s(p, k) + \sum_{s'_{inj}} mw_r(s'_{inj}, s_{inj}, p) cw^{out}(s'_{inj}, k, p) \quad \forall p \in P, s_{inj}, s'_{inj} \in S_{inj} \tag{3.52}$$

$$mw^{out}(s_{inj}, p) cw^{out}(s_{inj}, k, p) = mw^{in}(s_{inj}, p) cw^{in}(s_{inj}, k, p) + m(s_{inj}, k, p) \quad \forall p \in P, s_{inj} \in S_{inj} \tag{3.53}$$

Storage tank constraints

Constraints 3.54 and 3.55 determine the amount of reusable water in the storage tank at the first time slot and any other time slot, respectively. Constraint 3.56 states that the amount of water in the tank must never exceed that maximum allowable amount. Constraint 3.57 ensures that the reusable water storage tank is empty at the last time point.

$$q_s(p) = Q_s^{in} - \sum_{s_{inj}} mw_s^{out}(s_{inj}, p), \forall p \in P, p = 1, s_{inj} \in S_{inj} \qquad 3.54$$

$$q_s(p) = q_s(p-1) - \sum_{s_{inj}} mw_s^{out}(s_{inj}, p) + \sum_{s_{inj}} mw_s^{in}(s_{inj}, p-1) \qquad 3.55$$
$$\forall p \in P, s_{inj} \in S_{inj}$$

$$q_s(p) \leq Q_s^U, \forall p \in P \qquad 3.56$$

$$q_s(p) = 0, \forall p = |P| \qquad 3.57$$

Storage contaminant balance constraints

Constraints 3.58 and 3.59 determine the contaminant concentration of the reusable water in the storage tank in the first time slot and any other time slot, respectively. Constraint 3.59 considers the reusable water in the stank as well as the water that entered the tank from the previous time slot.

$$c_s(p,k) = C_s^{in}(k), \forall p = 1 \qquad 3.58$$

$$c_s(p,k) = \frac{q_s(p-1)c_s(p-1,k) + \sum_{s_{in,j}} mw_s^{in}(s_{inj}, p-1) cw^{out}(s_{inj}, k, p-1)}{q_s(p-1) + \sum_{s_{inj}} mw_s^{in}(s_{inj}, p-1)} \qquad 3.59$$
$$\forall p \in P, s_{inj} \in S_{inj}$$

Sequencing constraints

Constraints 3.32 to 3.37 still hold for this scenario. Constraint 3.60 and 3.61 ensure that the starting time of the washing operation receiving water coincide with the finishing time of the washing operations generating the wastewater being reused.

$$tw^{in}(s_{inj}, p) \geq tw^{out}(s'_{inj}, p) - H\left(1 - y_r(s_{inj}, s'_{inj}, p)\right) \quad \forall p \in P, s_{inj}, s'_{inj} \in S_{inj} \tag{3.60}$$

$$tw^{in}(s_{inj}, p) \leq tw^{out}(s'_{inj}, p) + H\left(1 - y_r(s_{inj}, s'_{inj}, p)\right) \quad \forall p \in P, s_{inj}, s'_{inj} \in S_{inj} \tag{3.61}$$

Constraints 3.62 and 3.63 ensure that the time at which a stream is transferred to the reusable storage coincides with the finishing time of the washing operation. Constraints 3.65 and 3.66 ensure that the time at which a stream is transferred from the reusable storage coincide with the starting time of the receiving washing operation. Constraints 3.64 and 3.67 ensure that for water to be transferred to and from storage, respective washing operations should be active in the same time slots.

$$t_s^{in}(s_{inj}, p) \geq tw^{out}(s_{inj}, p) - H\left(1 - y_s^{in}(s_{inj}, p)\right), \quad \forall p \in P, s_{inj} \in S_{inj} \tag{3.62}$$

$$t_s^{in}(s_{inj}, p) \leq tw^{out}(s_{inj}, p) + H\left(1 - y_s^{in}(s_{inj}, p)\right), \forall p \in P, s_{inj} \in S_{inj} \tag{3.63}$$

$$y_s^{in}(s_{inj}, p) \leq y(s_{inj}, p), \forall p \in P, s_{inj} \in S_{inj} \tag{3.64}$$

$$t_s^{out}(s_{inj}, p) \geq tw^{in}(s_{inj}, p) - H\left(1 - y_s^{out}(s_{inj}, p)\right), \quad \forall p \in P, s_{inj} \in S_{inj} \tag{3.65}$$

$$t_s^{out}(s_{inj},p) \le tw^{in}(s_{inj},p) + H\left(1 - y_s^{out}(s_{inj},p)\right), \forall p \in P, s_{inj} \in S_{inj} \qquad 3.66$$

$$y_s^{out}(s_{inj},p) \le y(s_{inj},p), \forall p \in P, s_{inj} \in S_{inj} \qquad 3.67$$

Constraint 3.68 states that the time at which water is transferred from the reusable water storage tank to a washing operation at any given time slot is later than the time at which water was transferred at the previous time slot. Constraint 3.69 is similar but applies to water transferred to a storage tank, i.e. it ensures that the time at which water is transferred from a unit to the reusable water storage tank at a time slot is later than the time at which water was transferred at a previous time slot. Constraint 3.70 ensures that the time at which water is transferred from the reusable water storage tank at a time slot is later than the time at which water was transferred to the reusable water tank at a previous time slot.

$$t_s^{out}(s_{inj},p) \ge t_s^{out}(s_{inj'},p-1) - H\left(2 - y_s^{out}(s_{inj},p) - y_s^{out}(s_{inj'},p-1)\right) \quad \forall p \in P, s_{inj}, s_{inj'} \in S_{inj} \qquad 3.68$$

$$t_s^{in}(s_{inj},p) \ge t_s^{in}(s_{inj'},p-1) - H\left(2 - y_s^{in}(s_{inj},p) - y_s^{in}(s_{inj'},p-1)\right) \quad \forall p \in P, s_{inj}, s_{inj'} \in S_{inj} \qquad 3.69$$

$$t_s^{out}(s_{inj},p) \ge t_s^{in}(s_{inj'},p-1) - H\left(2 - y_s^{out}(s_{inj},p) - y_s^{in}(s_{inj'},p-1)\right) \quad \forall p \in P, s_{inj}, s_{inj'} \in S_{inj} \qquad 3.70$$

Constraints 3.71 and 3.72 work together to ensure that the water transferred from the reusable tank to different washing operations in different units at the same time slot is transferred at the same time. Constraints 3.73 and 3.74 work together in a similar way but for water transferred from different washing operations to the reusable water tank in the same time slot.

$$t_s^{out}(s_{inj},p) \geq t_s^{out}(s_{inj'},p) - H\left(2 - y_s^{out}(s_{inj},p) - y_s^{out}(s_{inj'},p)\right) \quad \forall p \in P, s_{inj}, s_{inj'} \in S_{inj} \tag{3.71}$$

$$t_s^{out}(s_{inj},p) \leq t_s^{out}(s_{inj'},p) + H\left(2 - y_s^{out}(s_{inj},p) - y_s^{out}(s_{inj'},p)\right) \quad \forall p \in P, s_{inj}, s_{inj'} \in S_{inj} \tag{3.72}$$

$$t_s^{in}(s_{inj},p) \geq t_s^{in}(s_{inj'},p) - H\left(2 - y_s^{in}(s_{inj},p) - y_s^{in}(s_{inj'},p)\right) \quad \forall p \in P, s_{inj}, s_{inj'} \in S_{inj} \tag{3.73}$$

$$t_s^{in}(s_{inj},p) \geq t_s^{in}(s_{inj'},p) - H\left(2 - y_s^{in}(s_{inj},p) - y_s^{in}(s_{inj'},p)\right) \quad \forall p \in P, s_{inj}, s_{inj'} \in S_{inj} \tag{3.74}$$

3.6.5. Scenario 4: Fixed outlet concentration with sequence dependent changeover constraints and water reuse and recycle opportunity

The constraints for this scenario are based on both superstructures in Figures 3.1 and 3.2.Storage constraints 3.54 to 3.59 and sequencing constraints 3.32 to 3.37 and 3.60 to 3.74 hold for this scenario.

Water balance

In this scenario, constraints 3.39 to 3.41 and 3.47 to 3.51 still hold. Constraint 3.46 is replaced with 3.75 to ensure that freshwater, reused water, and water from the storage tank is utilized in stage A of the washing operation.

$$mw_a^{in}(s_{inj},p) = mw_f(s_{inj},p) + mw_s^{out}(s_{inj},p) + \sum_{s'_{inj}} mw_r(s'_{inj},s_{inj},p) \qquad 3.75$$
$$\forall p \in P, s_{inj}, s'_{inj} \in S_{inj}$$

Contaminant balance

Contaminant balance constraints 3.29, 3.44 and 3.45 still hold in this scenario. Constraint 3.76 replaces 3.43 by ensuring that the contaminant load in the outlet stream from stage A includes both the mass the load in the unit and the load in the inlet stream determined by constraint 3.77.

$$mw_a^{out}(s_{inj},p)cw_a^{out}(s_{inj},p) = mw_a^{in}(s_{inj},p)cw_a^{in}(s_{inj},p) + m(s_{inj},p) \qquad 3.76$$
$$\forall p \in P, s_{inj} \in S_{inj}$$

$$mw_a^{in}(s_{inj},p)cw_a^{in}(s_{inj},p) = mw_s^{out}(s_{inj},p)c_s(p)$$
$$+ \sum_{s'_{inj}} mw_r(s'_{inj},s_{inj},p)cw^{out}(s'_{inj},p) \qquad 3.77$$
$$\forall p \in P, s_{inj}, s'_{inj} \in S_{inj}$$

Tightening constraint

To tighten the model, constraint 3.73 ensures that every activity occurs within the time horizon of interest.

$$\sum_{s_{inj}} \sum_{p} \left(\begin{array}{l} \tau(s_{inj})y(s_{inj},p) + \beta(s_{inj})mu(s_{inj},p) \\ + tw_d(s_{inj},p)\left(\sum_{s_{inj}} c(s_{inj},s'_{inj},p+1,p) + a(s_{inj},p) \right) \end{array} \right) \le H \qquad 3.78$$
$$\forall p \in P, s_{inj} \in S_{inj}$$

3.6.6. Objective function

The objective is to maximize the profitability of a batch plant over the stipulated time horizon of interest. The objective function is made up of three components i.e. revenue, the cost of freshwater and cost of wastewater. However, each of the three components qualifies to be an objective function on its own.

Any of the objective functions, 3.79 to 3.82, can be chosen for any of the explored scenarios. Objective 3.79 maximizes revenue, 3.80 minimizes the cost of freshwater, 3.81 minimizes the cost of disposing of the effluent, and 3.82 maximizes profit, i.e. revenue minus water costs.

$$\text{Revenue} = \sum_{s_p \in S_p} q_s(s_p, p) SP(s_p) \qquad 3.79$$

$$\text{Cost of freshwater} = CFW \sum_{s_{inj} \in S_{inj}} \sum_{p \in P} mw_f(s_{inj}, p) \qquad 3.80$$

$$\text{Cost of Effluent} = CFE \sum_{s_{inj} \in S_{inj}} \sum_{p \in P} mw_e(s_{inj}, p) \qquad 3.81$$

$$\text{Objective} = \text{Revenue} - \text{Cost of Freshwater} - \text{Cost of Effluent} \qquad 3.82$$

3.8. References

Gouws, J.F., Majozi, T., Gadalla, M., 2008. Flexible mass transfer model for water minimization in batch plants. Chem. Eng. Process. Process Intensif. 47, 2323–2335. doi:10.1016/j.cep.2008.01.008

Majozi, T., Zhu, X.X., 2001. Novel continuous-time MILP formulation for multipurpose batch plants. 1. Short-term scheduling. Ind. Eng. Chem. Res. 40, 5935. doi:10.1021/ie0005452

Maranas, C.D., Zomorrodi, A.R., 2016. Optimization Methods in Metabolic Networks. Wiley.

Seid, R., Majozi, T., 2012. A robust mathematical formulation for multipurpose batch plants. Chem. Eng. Sci. 68, 36–53. doi:10.1016/j.ces.2011.08.050

Chapter 4

MODEL VALIDATION

4.1. Introduction

This chapter demonstrates the applicability of the formulations developed in the previous chapter. Two single contaminant illustrative examples and a multiple contaminant problem were examined. The two case studies were respectively presented by Kondili et al. (1993) and Maravelias and Grossmann (2003) for short-term scheduling of batch processes. In this work, the case studies have been adopted for wastewater minimization where sequence dependent water saving opportunities are explored. The illustrative example of Kondili et al. (1993) was also adopted for a multiple contaminant problem.

All four scenarios presented in the previous section are observed in both case studies. Scenarios 1 and 3 were validated separately from scenarios 2 and 4 since they require different sequence dependent changeover parameters, i.e. sequence dependent changeover washing water requirement and sequence dependent changeover rinsing fraction, respectively. Results were compared with a base case where no water saving opportunity was explored.

The resultant MINLP formulations were solved using a BARON solver in GAMS 24.3.2 in a computer with the following specifications: Windows 7 Professional, Intel(R) Core ™ i7.4770 CPU @ 3.40GHz, 8.00 GB RAM, and 64-bit Operating

System. Results are summarized in tables and production schedules are illustrated in Gant Charts. Objective function 3.82 was used for all scenarios in the formulations.

4.2. Illustrative example 1

4.2.1. Illustrative example description

The production recipe presented in Figure 4.1 shows that two chemical products, Product 1 and 2, are produced from three raw materials; Feed A, B and C. The production facility consists of four process units; a heater, two multipurpose reactors, and a separator. Both reactors (R) are suitable for processing reactions (Rxn) 1, 2 and 3.

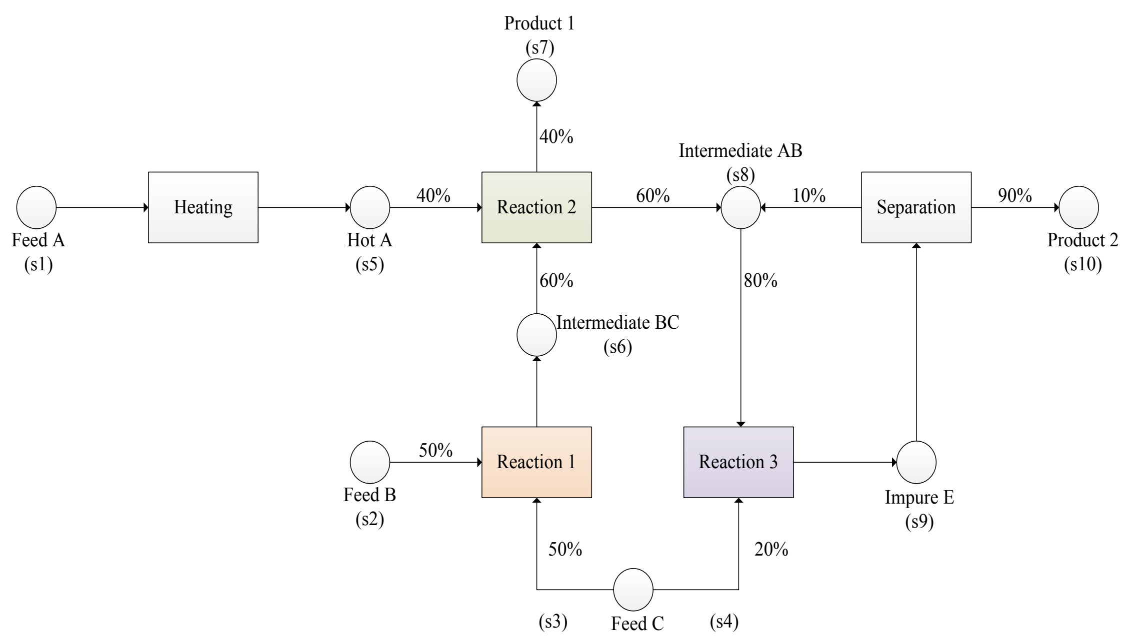


Figure 4.1 STN representation of the first illustrative example

A superstructure is a representation of all possible solutions. The superstructure in Figure 4.2 shows all possible sequences of tasks that can occur in both reactors 1 and 2. The mathematical model will, therefore, synthesize an optimal sequence of tasks for both reactors, which will be a subset of Figure 4.2.

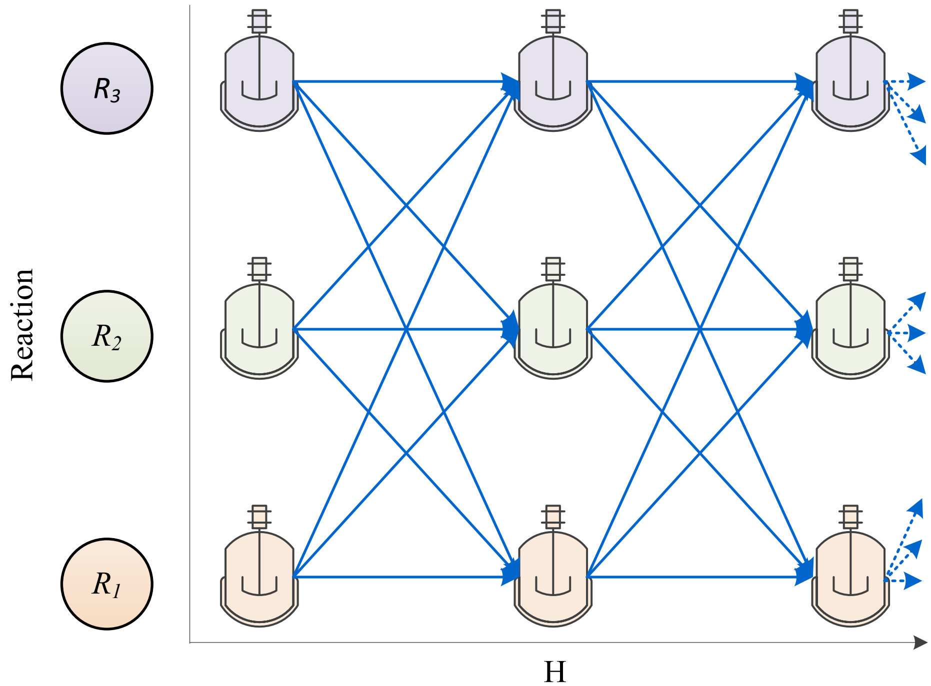


Figure 4.2 Sequence dependent changeover superstructure for the first illustrative example

The scheduling parameters for the Illustrative example are presented in Table 4.1. This includes the capacity of the available processing units which can be used as the upper bounds of the amount of material to be processed in that unit. The duration of processing task has a fixed term and a variable term which is influenced by the batch size. This means that the bigger the batch, the longer it will take to process it.

Table 4.1 Scheduling parameters for the first illustrative example

Unit	Unit capacity (kg)	Task	Effective states	α_{ij} (hr)	β_{ij} (h/kg)
Heater	100	H	S1	0.667	0.007
Reactor 1(R1)	50	Rxn1	S21	1.334	0.027
		Rxn2	S61	1.334	0.017
		Rxn3	S81	0.667	0.013
Reactor 2 (R2)	80	Rxn1	S22	1.334	0.027
		Rxn2	S62	1.334	0.017
		Rxn3	S82	0.667	0.008
Separator	200	Sr	S9	1.334	0.007

Contaminant concentration parameters for scenarios 1 and 3 as well as 2 and 4 are presented separately in Table 4.2. Contaminant concentration of the water for cleaning a unit after processing a task must not exceed the maximum inlet concentration and the water leaving a unit after the washing operation must not exceed the maximum outlet concentration.

Table 4.2 Maximum allowable inlet and outlet water concentration, illustrative example 1

Task (Symbol)	Scenario 1 and 3		Scenario 2 and 4	
	Max inlet conc. (g/kg)	Max outlet conc. (g/kg)	Max inlet conc. (g/kg)	Max outlet conc. (g/kg)
R1Rxn1	0.5	1	0.3	0.7
R1Rxn2	0.01	0.2	0.3	0.7
R1Rxn3	0.15	0.3	0.7	1.2
R2Rxn1	0.05	0.1	0.7	1.2
R2Rxn2	0.03	0.075	0.5	0.8
R2Rxn3	0.3	2	0.5	0.8

Table 4.3 presents other important parameters required in the modeling of illustrative example 1. This information that must be pre-determined include the time horizon of interest (H), the concentration of processed material that remain in the process unit (L), the selling price of products (SP), the cost associated with both freshwater (Cf) and wastewater (Cw), and the flowrate of the pressure cleaner (Rt).

Table 4.3 Other important parameters, illustrative example 1

Parameter	**Value**
H (hr)	14
L (g/kg)	1.2
SP_1 (c.u./kg)	20
SP_2 (c.u./kg)	20
W_f^{cost} (c.u/kg)	0.1
W_e^{cost} (c.u/kg)	0.05
Rt (kg/hr)	1200

Table 4.4 presents the sequence dependent parameters for scenarios 1 and 3, where the washing water requirement is fixed. For example, 140kg of water will be required for the cleaning in place washing operation if reaction 2 follows reaction 1 in reactor 1, and 110kg will be required if reaction 1 follows reaction 2 in the same reactor. It is assumed that the amounts specified in Table 4.4 are enough to remove the load and rinse the unit, the resultant outlet concentration will then be determined by the model even though it will not exceed the maximum outlet concentrations specified in Table 4.2. Sequence dependent washing requirement parameters for reactors 1 and 2 are different since these reactors have different capacities according to Table 4.1.

Table 4.4 Sequence dependent changeover washing water requirement in kilograms for scenarios 1 and 3, illustrative example 1

	R1Rxn1	**R1Rxn2**	**R1Rxn3**
R1Rxn1	-	140	160
R1Rxn2	110	-	130
R1Rxn3	210	190	-
	R2Rxn1	**R2Rxn2**	**R2Rxn3**
R2Rxn1	-	260	240
R2Rxn2	200	-	180
R2Rxn3	330	310	-

Table 4.5 presents the sequence dependent parameters that apply to both reactors for scenario 2 and 4, i.e. the additional fraction of the amount used for cleaning that will be used for rinsing. In scenarios 2 and 4, the contaminant concentration of the wastewater generated from the washing operations is fixed to the maximum outlet concentrations specified in Table 4.2. The model then determines the amount of water required to remove the load. Values specified in Table 4.5 are the percentage of the amount required for washing that must be used for rinsing the unit. For example, 85% of the water used for rinsing reactor 1 or 2 must be used for rinsing if reaction 2 follows reaction 1; and 75% must be used if reaction 1 follows reaction 2.

Table 4.5 Sequence dependent rinsing fraction for scenarios 2 and 4, illustrative example 1

	Reaction 1	Reaction 2	Reaction 3
Reaction 1	-	0.85	0.45
Reaction 2	0.75	-	0.95
Reaction 3	0.80	0.60	-

4.2.2. Results

Results for all scenarios are summarized in Tables 4.6 and 4.7 and graphically presented in Gantt Charts (Figures 4.3 to 4.7). In the Gantt Charts, the available units are on the vertical axis and the time horizon of interest is on the horizontal axis. Blocks with texts represent the task that occurred in the unit and the amount processed is written in brackets. Blocks with no texts represent washing operations and the amount of water required is also presented in kilograms.

a. Scenarios 1 and 3

Table 4.6 summarizes the results for scenarios 1 and 3 where the sequence dependent washing requirement was a parameter. The Gantt Charts for the base case, scenario 1 and scenario 3 are presented in Figures 4.3 to 4.5 respectively. These charts present information such as the production schedule, water requirement, duration of washing and the water network. When direct and indirect water reuse and recycle opportunities for water minimization were explored alone, 34% of the water required in the base case was saved. Scenarios 1 and 3 saved 53% and 66% of the total amount required by the base case respectively. Scenario 1 saved water by simply synthesizing a sequence of tasks that optimizes the trade-off between production and wastewater minimization.

Table 4.6 Results for scenarios 1 and 3, illustrative example 1

	Base case	**Direct/indirect reuse**	**Scenario 1**	**Scenario 3**
Objective (c.u)	5570.75	5680.99	5713.07	5755.07
Water (kg)	2196	1434	1020	740
Water saved (%)	-	33.89	52.97	65.88
CPU time (sec)	7	4440	44	4380

Figure 4.3 is the Gantt Chart of the described problem when none of the water saving opportunities are explored. Multipurpose reactors are being washed after processing any task since the amount of water required for washing operation is determined only by the task that has just been processed in the unit

.

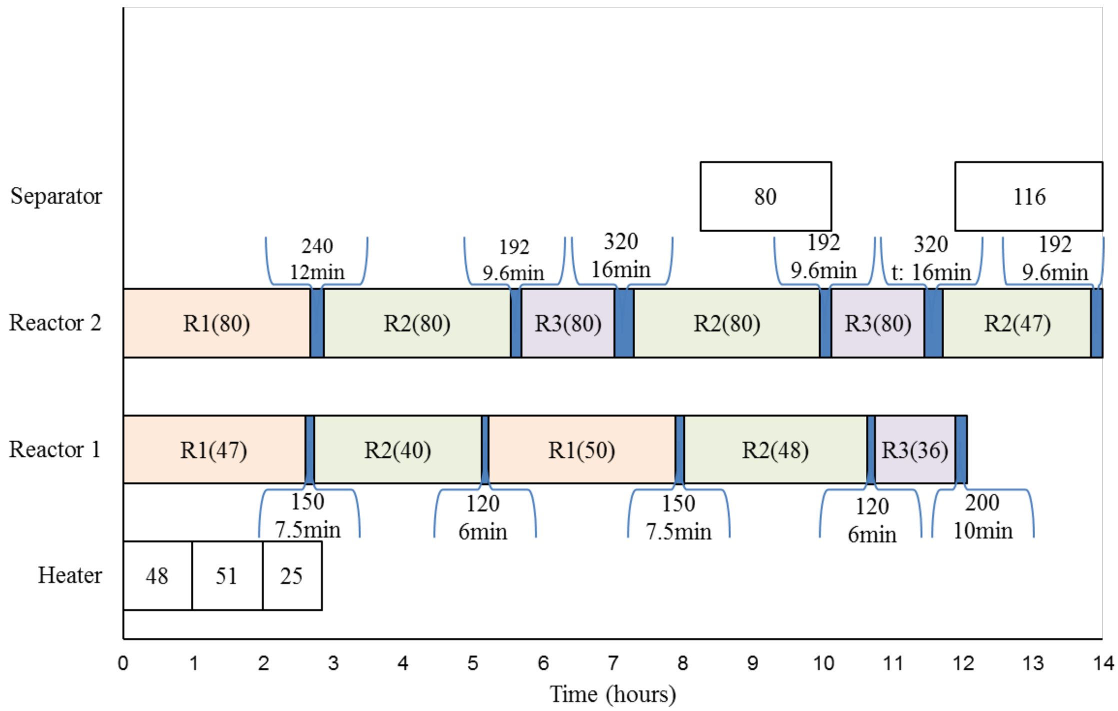


Figure 4.3 Base case for scenario 1 and 3, illustrative example 1

The Gantt chart in Figure 4.4 shows that scenario 1 favored the campaign mode or the processing of successive batches of the same task since it did not require changeover washing operations. Figure 4.4 also shows that reaction 3, which produces product 2, was only processed in reactor 1. This setup is favored such that the sequence of consecutive of the same task is maximized.

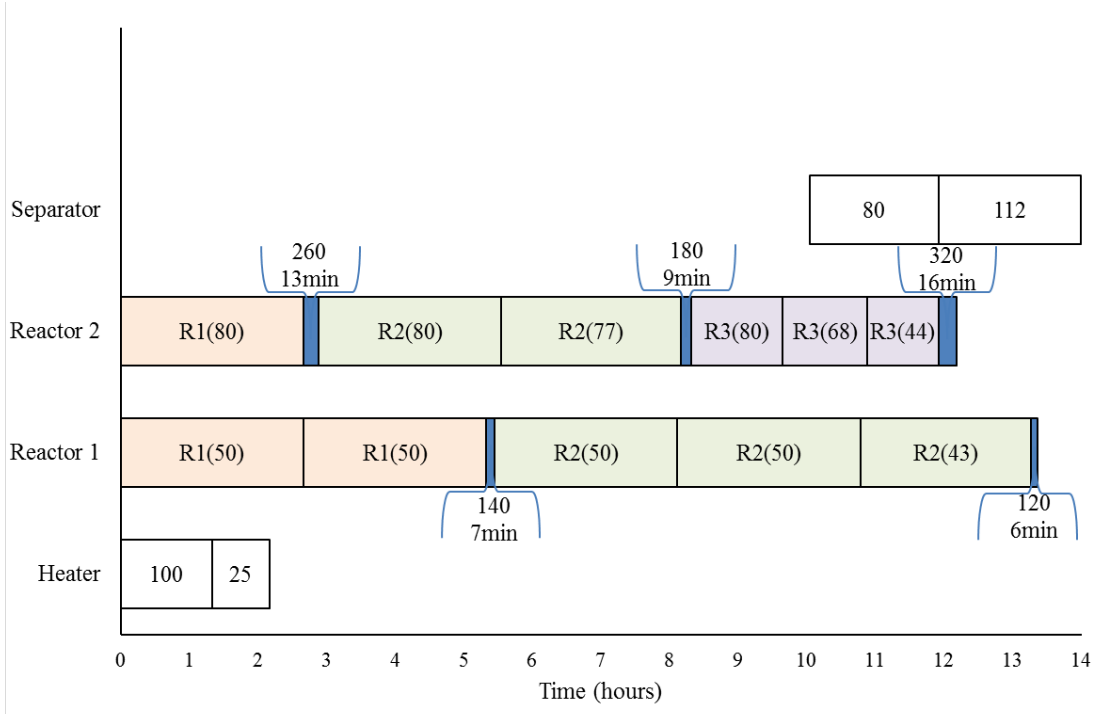


Figure 4.4 Scenario 1, illustrative example 1

The Gantt chart in Figure 4.5 shows that direct and indirect water reuse and recycle opportunities were found for scenario 3. Exploring multiple water saving opportunities, i.e. sequence dependent changeover opportunities with water reuse and recycle, resulted in even greater water savings. However, this also resulted in the increase in the computational time for solving the optimization program due to the increased model size and complexity.

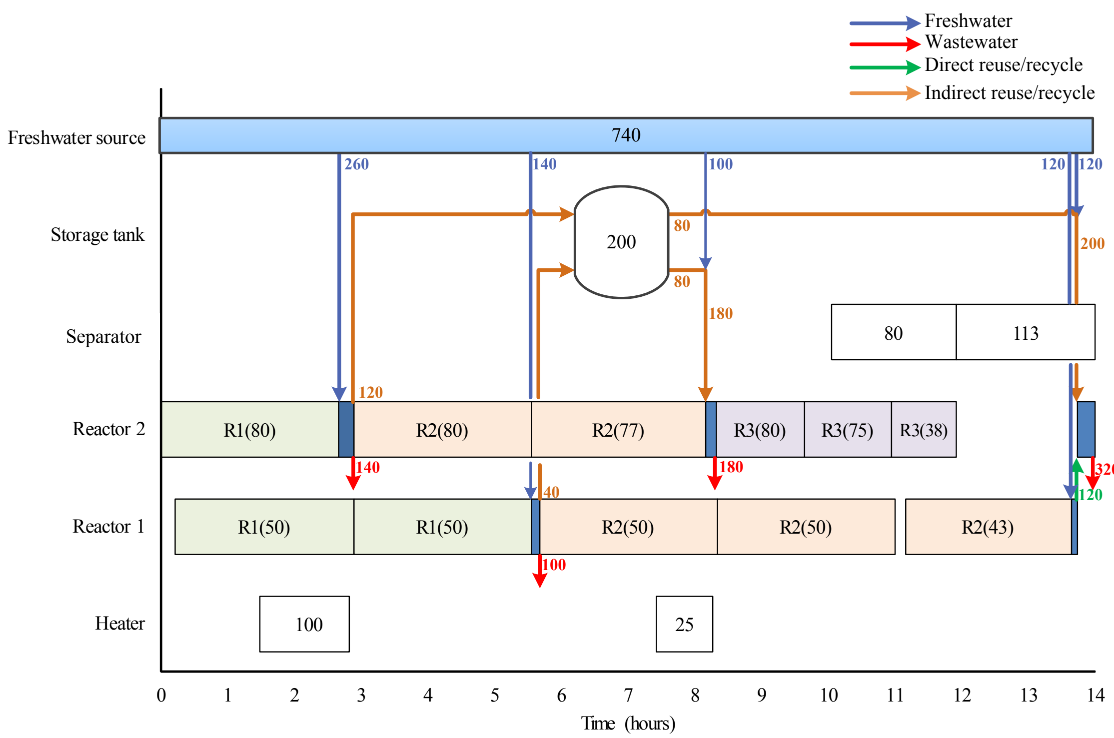


Figure 4.5 Scenario 3, illustrative example 1

b. Scenarios 2 and 4

Table 4.7 summarizes the results for scenarios 2 and 4 where the outlet contaminant concentration was fixed to a given maximum and a total amount of water for cleaning and rinsing during a washing operation was determined.

Scenario 2 synthesized a sequence of tasks that optimize production and wastewater generation based on the sequence dependent rinsing operations. Scenario 4 expanded scenario 2 by exploring both sequence dependent water saving opportunities with direct or indirect water reuse and recycle opportunities simultaneously. 13% and 45% of washing water required by the base case were respectively saved when water reuse and recycle and sequence dependent changeover opportunities for water minimization were explored separately.

Table 4.7 Results for scenarios 2 and 4, illustrative example 1

	Base case	**Direct/indirect reuse**	**Scenario 2**	**Scenario 4**
Objective (c.u)	4277.13	4341.75	4802.38	4907.30
Water (kg)	3125.65	2693.84	1632.42	1458.91
Water saved (%)	-	13.82	47.77	53.32
CPU time (sec)	5	5400	17	5400

The Gantt Charts for the production schedule of the base case is presented in Figure 4.6. The amount of water required for a washing operation depends only on the task that has just been processed in a unit. Water required by the base case can be reduced by 13.82% by exploring direct and indirect water reuse and recycle. In scenario 2, sequence dependent water saving opportunities were explored and 47.77% of water required for washing operations was saved (Figure 4.7). 53.32% was saved in scenario 4 where scenario 2 was expanded by simultaneously exploring direct or indirect water reuse and recycle opportunities (Figure 4.8).

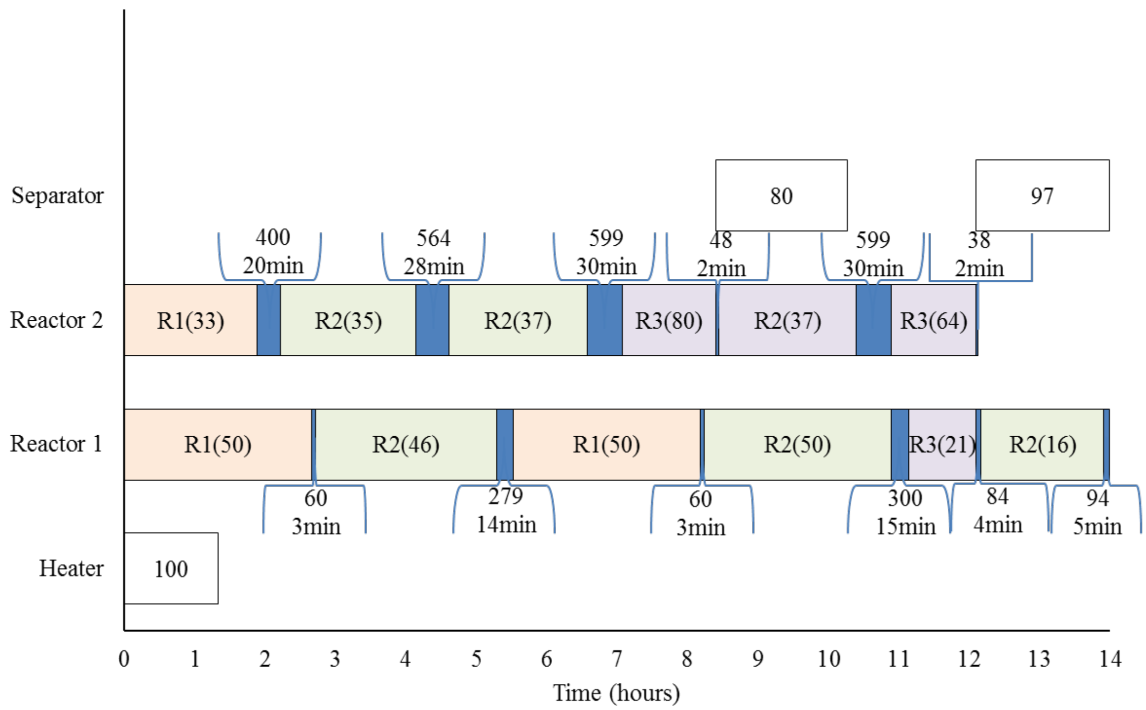


Figure 4.6 Base case for scenario 2 and 4, illustrative example 1

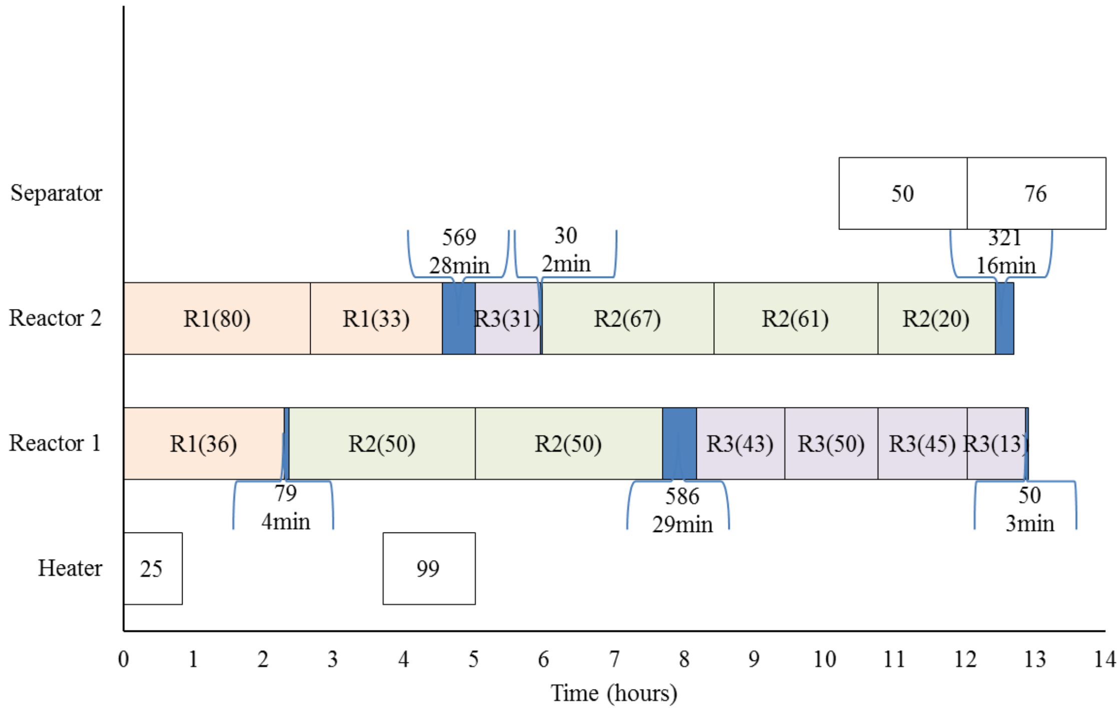


Figure 4.7 Scenario 2, illustrative example 1

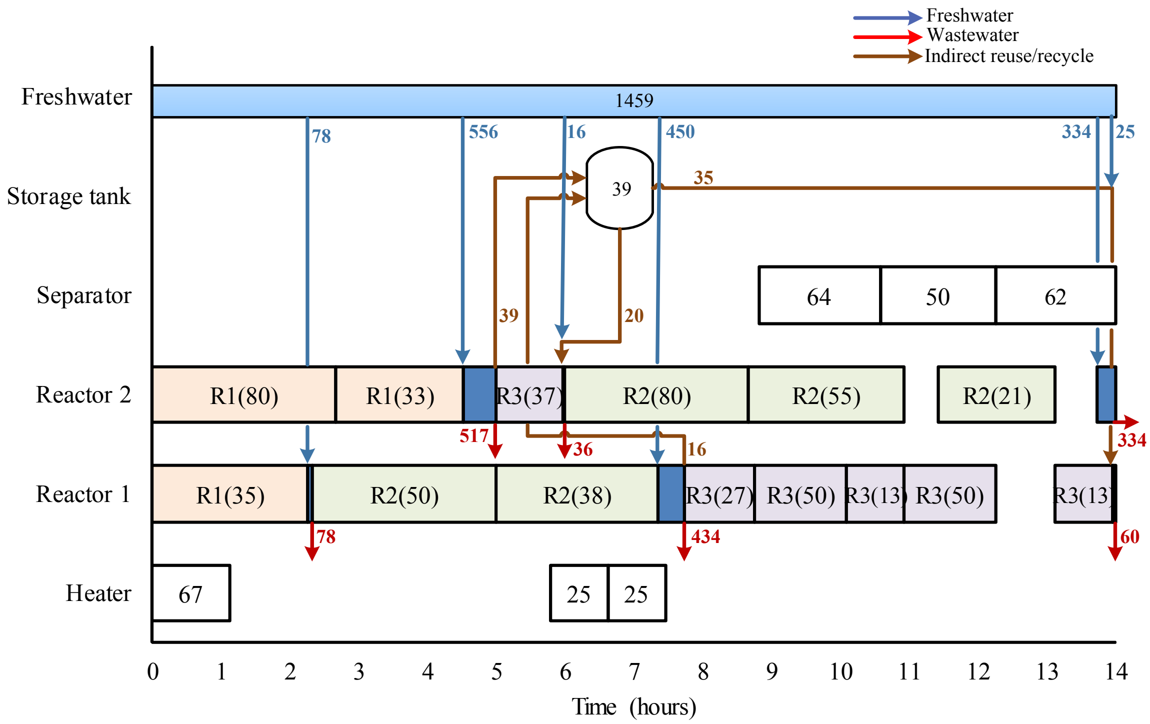


Figure 4.8 Scenario 4, illustrative example 1

4.3. Illustrative example 2

4.3.1. Illustrative example description

Figure 4.9 shows a production recipe where each of the two products, P1 and P2, is produced from one raw material and three tasks. The batch facility consists of two processing units and storage tanks for each state. The process unit U1 is suitable for processing task one of product one (T11), task one of product two (T21), task three of product one (T13) and task three of product two (T23). Task two of product one (T12) and task two of product two (T22) can be processed in processing unit U2.

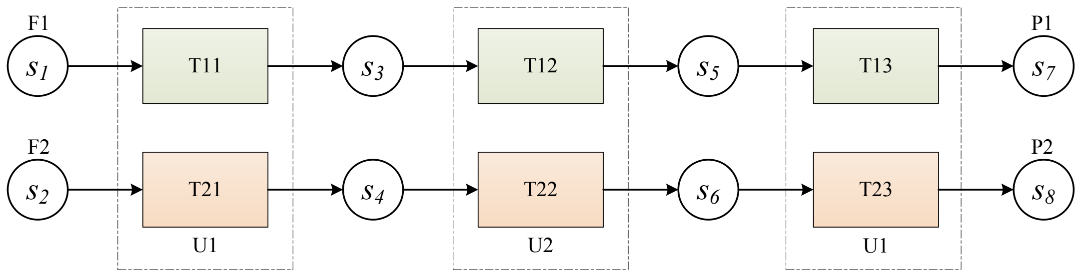


Figure 4.9 STN representation of the second illustrative example

Figure 4.10 represents two superstructures representing all possible sequence of tasks that could occur in units U1 and U2 respectively. The formulation should, therefore, synthesize an optimal sequence of tasks for each processing unit.

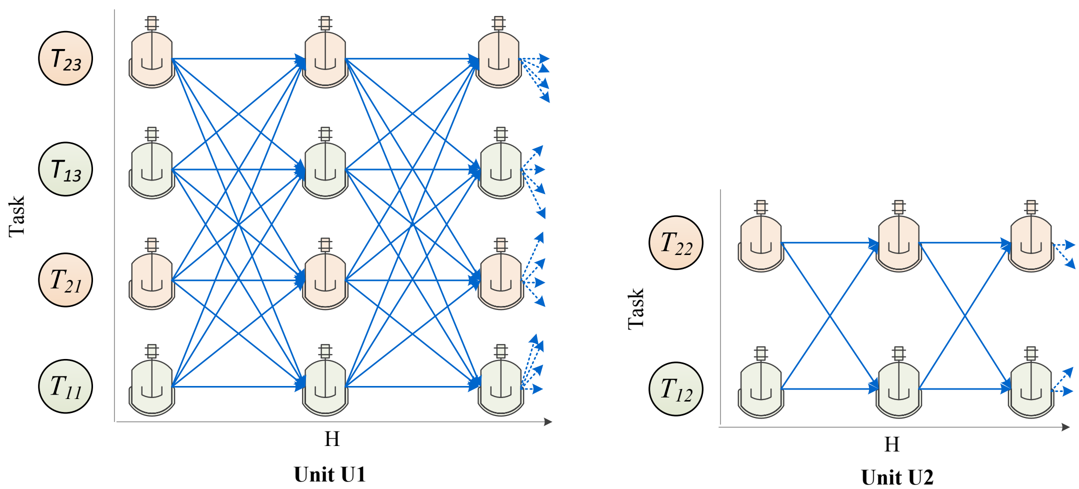


Figure 4.10 Sequence dependent changeover superstructures for the second illustrative example

Scheduling data, contaminant concentration limits and other relevant production data for the second literature example are respectively presented in Tables 4.8, 4. 9 and 4.10. Tables 4.11 and 4.12 presents the sequence dependent data, i.e. fixed water requirement (for scenarios 1 and 3) and fixed outlet concentration (for scenarios 2 and 4).

Table 4.8 Scheduling parameters for the second illustrative example

Unit	Min batch size (T)	Max batch size (T)	Task	Effective states	α_{ij} (hr)	β_{ij} (hr/t)
U1	2	5	T11	s1	0.5	0.40
			T21	s21	0.75	0.60
			T13	s61	0.5	0.40
			T23	s81	0.5	0.40
U2	1.2	3	T12	s22	1.0	1.33
			T22	s62	1.0	1.33

Table 4.9 Maximum allowable inlet ant outlet water concentration, illustrative example 2

Task	Max inlet concentration (Kg/T)	Max outlet concentration (Kg/T)
T11	0.5	1
T21	0.5	1
T13	1	2
T23	1	2
T12	0.5	1
T22	0.5	1

Table 4.10 Other important parameters, illustrative example 2

Parameter	**Value**
H (hr)	12
L (g/kg)	0.1
SP_1 (c.u./kg)	1
SP_2 (c.u./kg)	1
$W_f^{\cos t}$(c.u/kg)	0.5
$W_e^{\cos t}$ (c.u/kg)	0.25
Rt (kg/hr)	1200

Table 4.11 Sequence dependent changeover washing water requirement in kilograms for scenarios 1 and 3, illustrative example 2

	T11	**T21**	**T12**	**T22**	**T13**	**T23**
T11	0	350	-	-	300	310
T21	220	0	240	-	200	243
T12	-	-	0	117	-	-
T22	-	-	121	0	-	-
T13	300	320	-	-	0	340
T23	200	240	-	-	242	0

Table 4.12 Sequence dependent rinsing fractions for scenarios 2 and 4, illustrative example 2

	T11	**T21**	**T12**	**T22**	**T13**	**T23**
T11	0	0.8	-	-	0.6	0.7
T21	0.5	0	0.8	-	0.4	0.6
T12	-	-	0	0.7	-	-
T22	-	-	0.5	0	-	-
T13	0.4	0.5	-	-	0	0.8
T23	0.4	0.6	-	-	0.7	0

4.3.2. Results

a. Scenarios 1 and 3

Table 4.13 summarizes the outcomes for scenarios 1 and 3. A total amount of 2690kg of water was required by the base case as represented in Figure 4.11. Exploring direct and indirect opportunities resulted in 13% savings in freshwater. Sequence dependent changeover opportunities for water minimization explored in scenario 1 resulted in 49% water savings, and the Gantt Chart is presented in Figure 4.12. 61% of the water required in the base case is saved when sequence dependent opportunities for water minimization were explored simultaneously with direct and indirect water reuse and recycle (Figure 4.13).

Table 4.13 Results for scenarios 1 and 3, illustrative example 2

	Base case	Direct/indirect reuse	Scenario 1	Scenario 3
Objective (c.u)	7.223	7.474	8.522	8.687
Water (kg)	2690	2342	1372	1050
Water saved (%)	-	13	49	61
CPU time (sec)	12	5400	48	3790

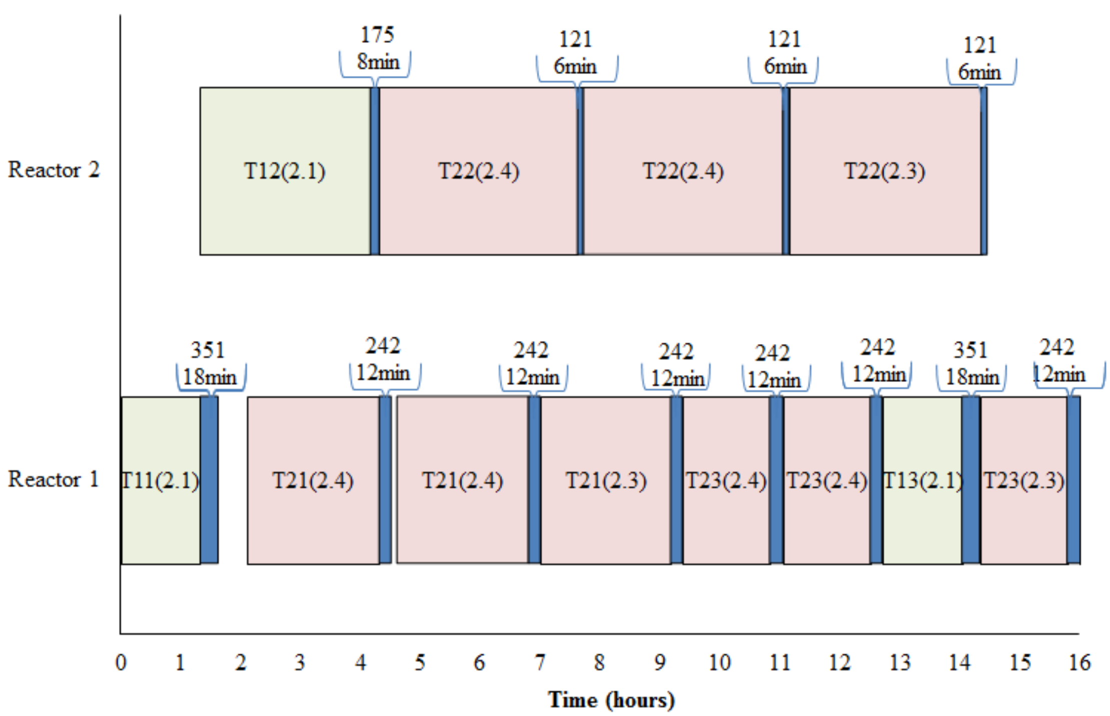


Figure 4.11 Base case for scenarios 1 and 3, illustrative example 2

When incorporating sequence dependent changeover constraints, the formulation favored the campaign mode, i.e. a sequence of similar batches of the same task. This resulted in significant water savings since fewer washing operations are required.

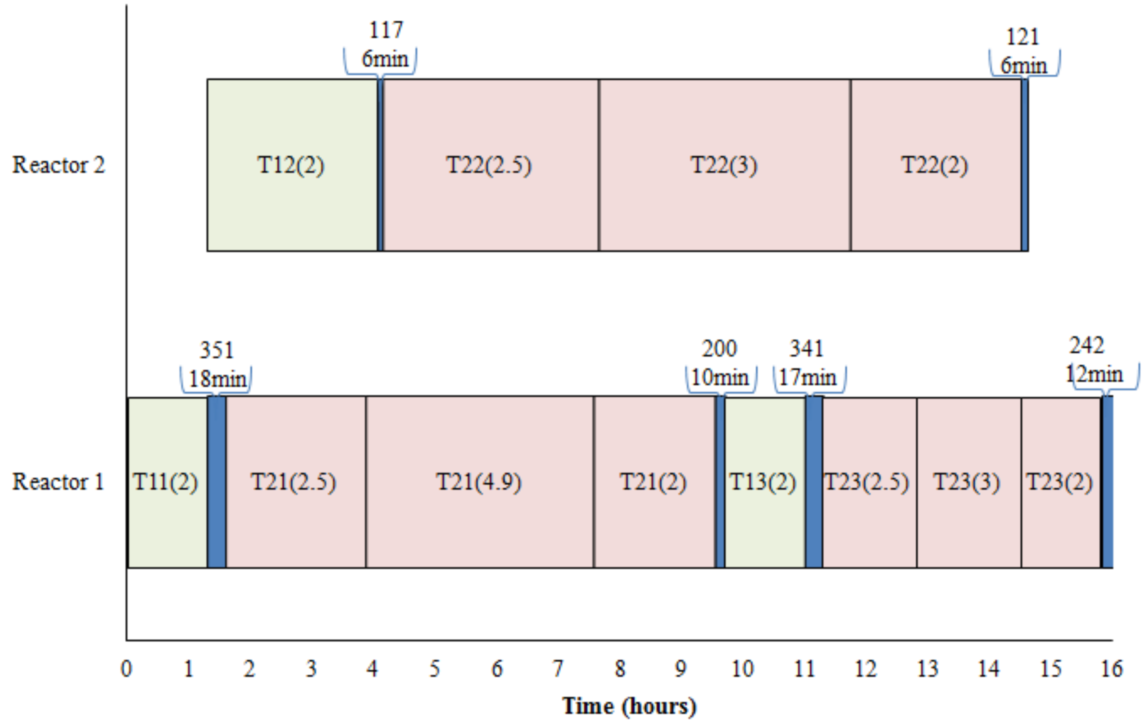


Figure 4.12 Scenario 1, illustrative example 2

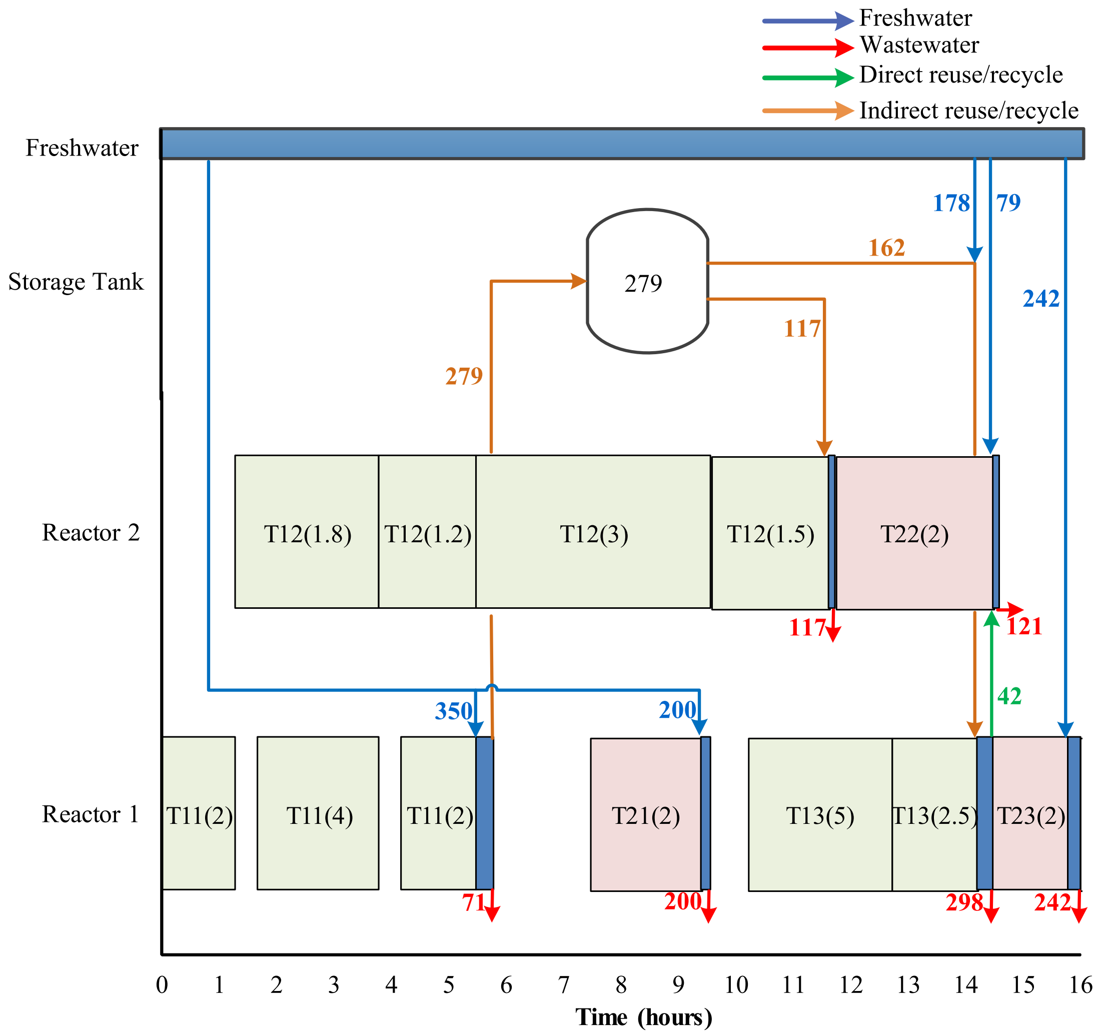


Figure 4.13 Scenario 3, illustrative example 2

b. Scenarios 2 and 4

Table 4.14 summarizes the results for scenarios 2 and 4. A total of 2358 kg of water was required by the base case which is represented by the Gantt Chart in Figure 4.14. Sequence dependent changeover opportunities for water minimization explored in scenario 2 resulted in 41% water savings, and the Gantt Chart is presented in Figure 4.15. This Gantt Chart also holds for scenario 4 since no reuse and recycle opportunities were found when sequence dependent opportunities were explored simultaneously with direct and indirect water reuse and recycle opportunities.

Table 4.14 Results for scenarios 2 and 4, illustrative example 2

	Base case	Direct/indirect reuse	Scenario 1	Scenario 3
Objective (c.u)	7.651	7.797	8.458	8.458
Water (kg)	2354	1933	1402	1402
Water saved (%)	.	18	41	41
CPU time (sec)	2	5400	105	5400

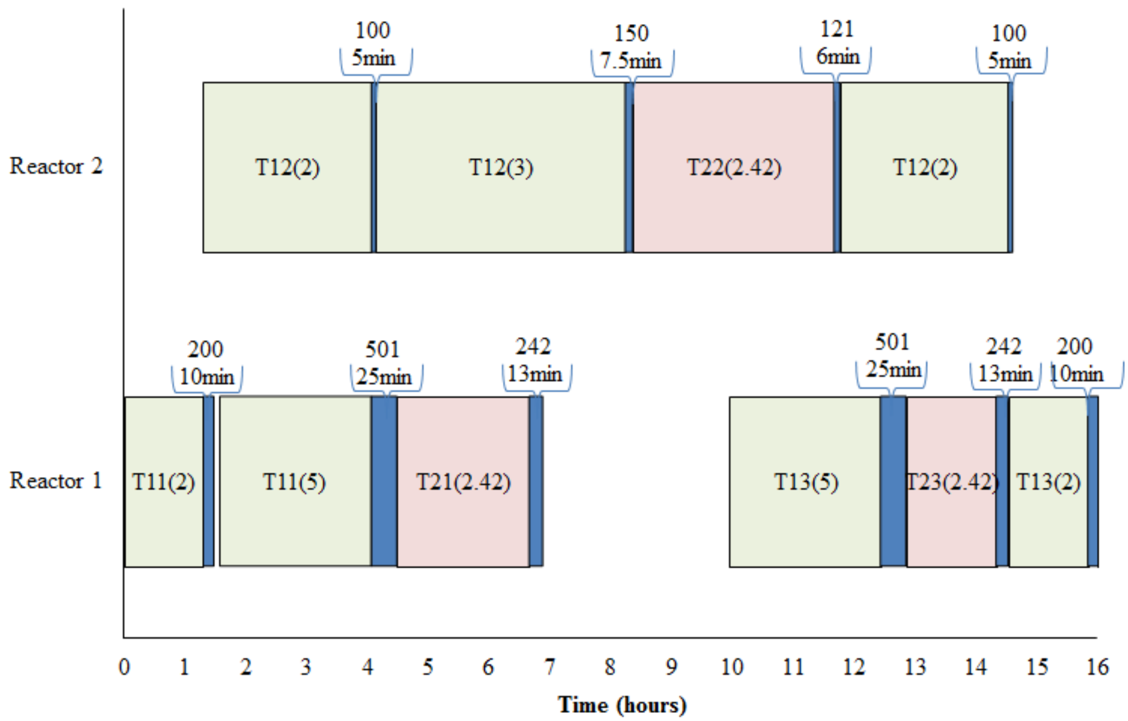


Figure 4.14 Base case for scenarios 2 and 4, illustrative example 2

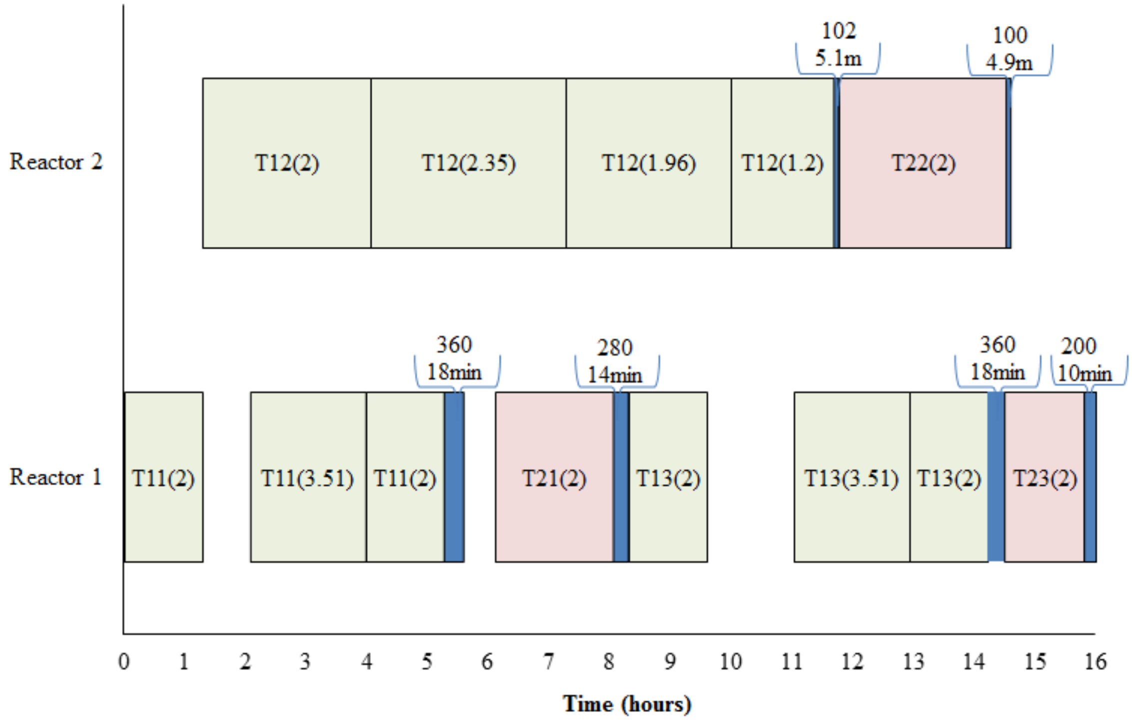


Figure 4.15 Scenario 2, illustrative example 2

4.4. Multiple contaminant example

An illustrative example, represented by an STN in Figure 5, was used to demonstrate the application of the developed formulation on multiple contaminant problems. The superstructure in Figure 6 and parameters in Table 4.1, 4.3 and 4.4 are still applicable. However, the concentration limits and mass load fractions for the multiple contaminant example are presented in Table 4.15 and 4.16 respectively. Using the multiple contaminant parameters, formulations for scenarios 1 and 3 were applied.

Table 4.15 Concentration limits for the multiple contaminants example

Task	Contaminant 1 (k1)		Contaminant 2 (k2)		Contaminant 3 (k3)	
	Max inlet conc. g/kg)	Max outlet conc. (g/kg)	Max inlet conc. (g/kg)	Max outlet conc. (g/kg)	Max inlet conc. (g/kg)	Max outlet conc. (g/kg)
R1Rxn1	0.5	1	0.5	0.9	2.3	3

R1Rxn2	0.01	0.2	0.05	0.1	0.3	1.2
R1Rxn3	0.15	0.3	0.2	1	0.35	1.2
R2Rxn1	0.05	0.1	0.2	1	0.05	1.2
R2Rxn2	0.03	0.075	0.1	0.2	0.2	1
R2Rxn3	0.3	2	0.6	1.5	1.5	2.5

Table 4.16 Mass load (L in g/kg) for the multiple contaminants example

	Contaminant 1 (k1)	Contaminant 2 (k2)	Contaminant 3 (k3)
R1Rxn1	0.051	1.021	0.126
R1Rxn2	0.045	0.072	1.082
R1Rxn3	0.200	0.052	0.947
R2Rxn1	0.400	0.089	0.711
R2Rxn2	0.100	0.533	0.567
R2Rxn3	0.259	0.519	0.421

Results of the multiple contaminant example are summarized in Table 4.17. Direct and indirect water reuse opportunities saved 13% of the water used by the base case, while scenarios 1 and 3 saved 31% and 42% respectively. Scenario 4 achieved higher water savings by exploring sequence dependent water saving opportunities and reuse and recycle opportunities, through a central storage tank, simultaneously. Formulations that explore reuse and recycle water saving opportunities have high nonlinearity and they required greater computational times. Figures 4.16, 4.17 and 4.18 are the Gantt Charts showing the graphical representation of the base case, scenario 1 and scenario 3, respectively.

Table 4.17 Results for scenario 1 and 3, multiple contaminants example

	Base case	Direct/indirect reuse	Scenario 1	Scenario 3
Objective (c.u)	4805.59	4854.942	5509.042	5542.642
Water (kg)	2526	2196.92	1732	1471.9

Water saved (%)	-	13.02	31.43	41.73
CPU time (s)	3	10000	18	10000

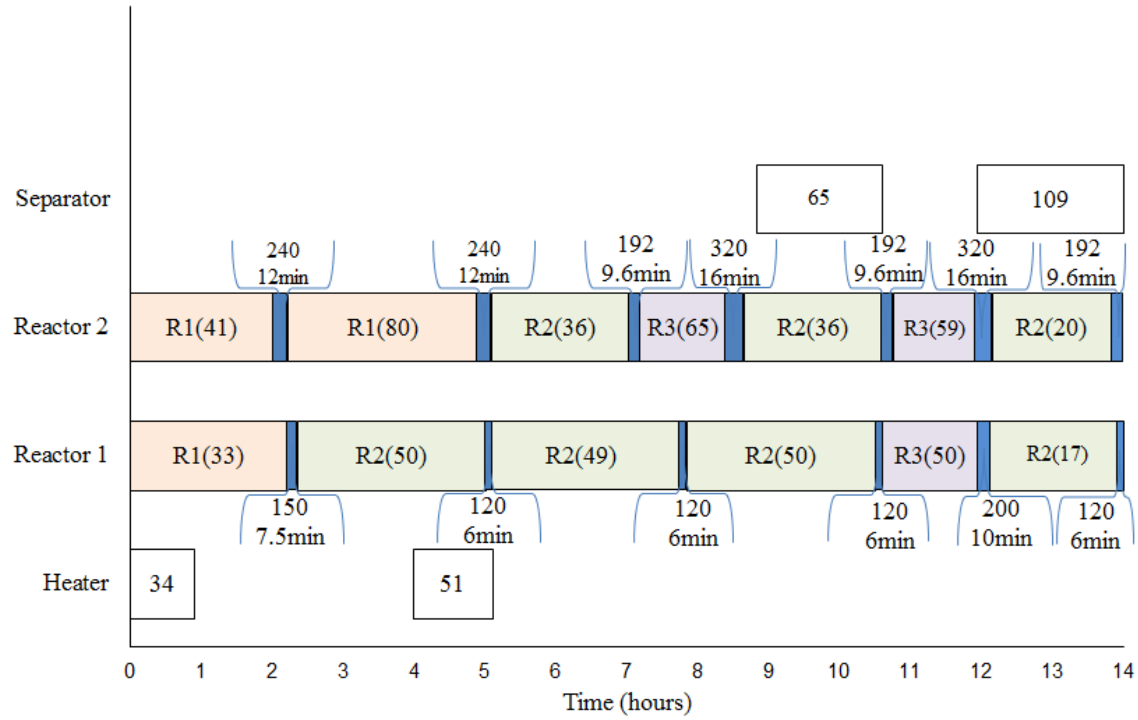


Figure 4.16 Base case, multiple contaminants example

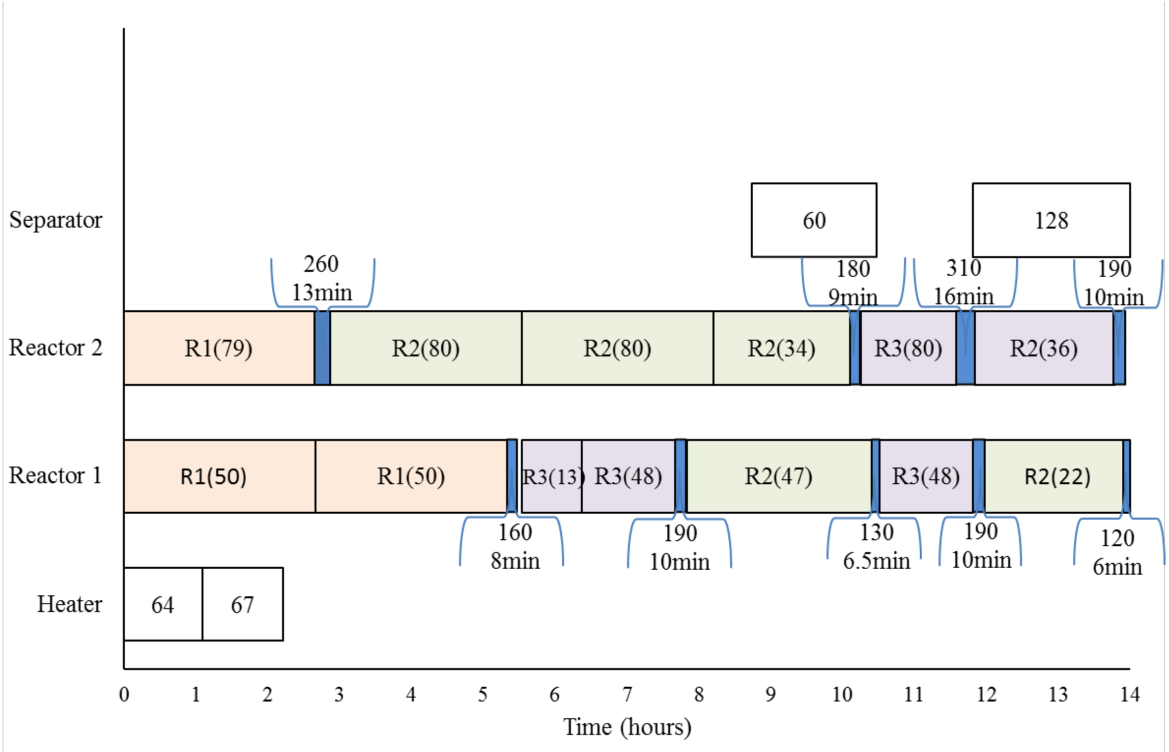


Figure 4.17 Scenario 1, multiple contaminants example

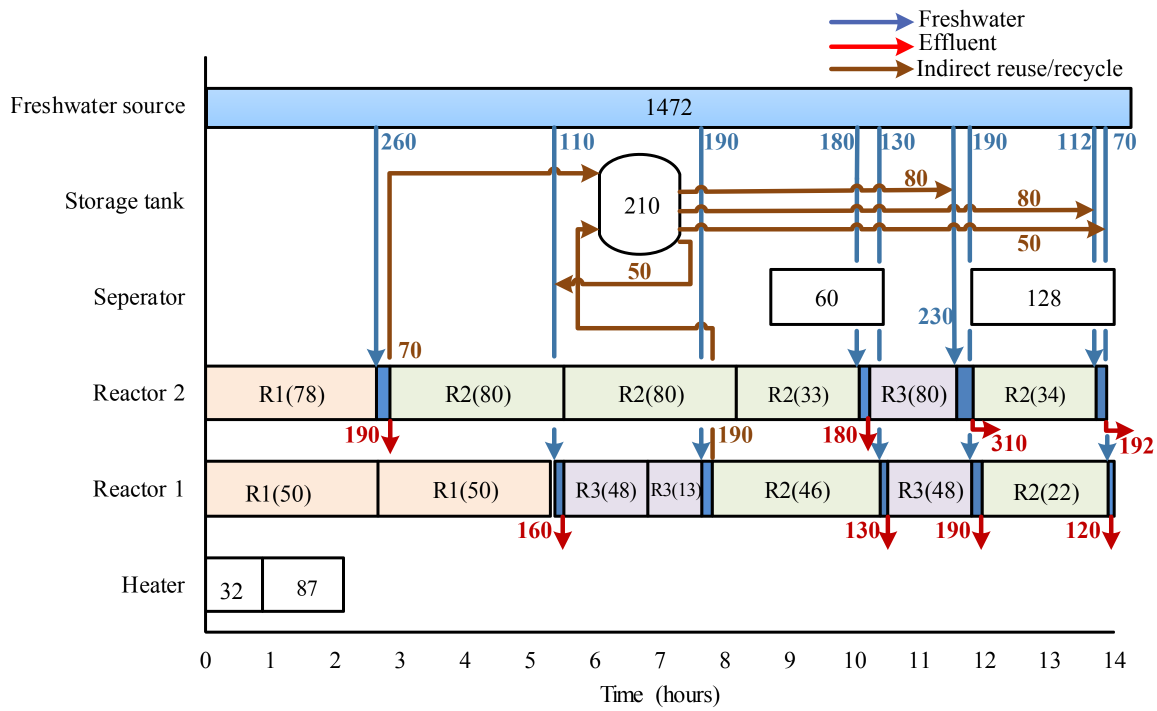


Figure 4.18 Scenario 3, multiple contaminants example

4.5. Discussion

Developed formulations were applied to two single contaminant problems, and a multiple contaminant problem for fixed water requirement scenarios. Higher percentages of water savings were achieved by scenarios that explored sequence-dependent water saving opportunities simultaneously with water reuse and recycle. The drawback, however, was that these scenarios result in more complex formulations and it can happen that the water reuse and recycle opportunities are not found, as observed in scenario 4 of illustrative example 2. Unfortunately, this can only be observed after the computationally intensive model has been solved. Indirect water reuse and recycle opportunities are often not found as a result of the contaminant concentration limits since they inform constraints that ensure that the concentration requirements are met.

When comparing scenarios that explore sequence-dependent water saving opportunities and water reuse and recycle techniques separately, the former saved more water and took less time to solve in both illustrative examples. Scenarios that explore sequence-dependent water saving opportunities saved more water by synthesizing a sequence of tasks that minimize the number of required washing operations. This was possible since the illustrative examples did not require washing between consecutive batches of the same task.

The size of the central water storage tank that will be required when exploring indirect water reuse and recycle opportunities was determined from the maximum amount of water that in the tank over the time horizon of interest. Any vessel or process unit that has a capacity equal to or greater than the one required, can be used to facilitate the indirect water reuse and recycle opportunities.

In the first illustrative example, a complex formulation for scenario 3 solved faster than a less complex model that only explored water reuse and recycle techniques. The more complex base case and scenario 3 of the multiple contaminant example solved faster than those of the less complex single contaminant illustrative examples 1 and 2. This is due to the size of the search space for an optimum solution. Solution algorithms and solvers, including BARON, take more time to find an optimum solution if the search space is bigger. Therefore, as much as the formulations for scenarios that explore sequence-dependent water saving opportunities simultaneously with water reuse and recycle are larger, they have a smaller search space compared to scenarios that only explore water reuse and recycle techniques. The reduced search space is due to additional constraints or imposed variable bounds.

The toxicity of the wastewater generated from batch processes is a major concern in batch manufacturing. The proposed formulation, however, is able to ensure that the contaminant concentration of the wastewater to be disposed to the environment does not exceed the maximum allowable concentration. The contaminant concentration upper limit is imposed in scenarios 1 and 3, and the contaminant concentration is fixed for scenarios 2 and 4.

The proposed formulations were validated using two single contaminant problems and a multiple contaminant problem for fixed water requirement scenarios. However, multiple contaminant problems with fixed contaminant concentration may exist. Formulations for fixed contaminant concentration presented in this work (scenarios 2 and 4) can only allow one contaminant concentration to be fixed. It is, however, possible to represent multiple contaminant problems in a way that allows them to be applied to single contaminant optimization models. Approaches to do this include identifying a key contaminant and finding an aggregate or average contaminant concentration.

4.6. References

Kondili, E., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations—II. Computational issues. Comput. Chem. Eng. 17, 229–244. doi:10.1016/0098-1354(93)80016-G

Maravelias, C.T., Grossmann, I.E., 2003. New General Continuous-Time State - Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants 3056–3074.

Chapter 5

LIMITATIONS AND RECOMMENDATIONS

5.1. Introduction

The proposed mathematical formulations demonstrated promising results for significant water savings in multipurpose batch plants. However, the presented model has limitations or shortcomings. Presented in this chapter are the limitations of the presented formulation as well as recommendations that might influence future research. Discussed issues include accounting for data collection challenges, computational time, and exploring other possible water saving opportunities.

5.2. Multiple contaminants

The proposed formulations were validated using two single contaminant problems and a multiple contaminant example for fixed water requirement scenarios. However, multiple contaminant problems with fixed contaminant concentration may exist. Formulations for fixed contaminant concentration presented in this work allows only one contaminant concentration to be fixed. It is, however, possible to represent multiple contaminant problems in a way that allows them to be applied to single contaminant optimization models. Approaches to do this include identifying a key contaminant and finding an aggregate or average contaminant concentration.

Identifying a key contaminant

This approach assumes that, from a stream with multiple contaminants, only one of them is of significant quantity or have a greater impact on the environment and other contaminants is negligible. This assumption is often justifiable in situations where one contaminant is in abundance relative to the others. After identifying the key contaminant, all parameters in the modeling will, therefore, be based on the identified key contaminant (Savelski and Bagajewicz, 2003).

This method has its drawbacks. Sometimes choosing the key contaminant is not a straightforward task. The key contaminant cannot be identified if the composition of the stream is unknown. Also, not accounting for contaminants that are assumed to have negligible effects can result in inaccuracies in the output of the model.

Average and aggregate contaminant concentration

This approach proposes a simple average of all contaminants involved in the stream. A more popular option that follows a similar thinking will be to group contaminants into aggregate properties such as total dissolved solids, total load, biochemical oxygen demand, chemical oxygen demand, etc. Both these approaches consider all contaminants involved, unlike the above approach that only considers one key contaminant. However, these approaches may still neglect the effect of individual contaminants.

This simplification of complex multiple contaminant problems can lead to models that do not accurately represent the real case or results that are impractical. Multiple contaminants are more prevalent in industry than single contaminants (Majozi and Gouws, 2009). Also, models formulated for multiple contaminant problems can be easily adapted to single contaminant problem than trying to apply a multiple contaminant problem to a model formulated for a single contaminant problem.

5.3. Model Validation

The quality of the input data is one of the very important factors that influence the reliability of the output of an optimization model. Other factors may include the relationship between variables, constraints, objective function, etc. The sequence dependent changeover data required by the proposed formulations, for the sequence dependent water saving opportunities to be explored, may be challenging to obtain. For example, the fixed sequence dependent water requirement can be obtained after a long investigation of observing cleaning in place operations and different batches of different tasks being processed in the same unit. This means that for these formulations to be implemented practically, significant efforts must be invested in trying to obtain the required data.

The developed data was validated using illustrative literature examples. This work can be extended further by validating the developed formulation using a real life industrial plant.

5.4. Computational intensity

The proposed formulations were successfully solved for the illustrative examples using the branch and reduce optimization navigator (BARON) solver. For some scenarios, however, the optimization model took too much time to find an optimum solution. This may prove to be a problem in facilities where production schedules need to be generated regularly. Factors influencing the computational time of an optimization model include the model complexity and the search space. It was observed that more complex models require larger computational times. The complexity of an optimization model can be increased by adding more constraints that consider more factors, by increasing the desired time horizon of interest, etc. Computational challenges may be addressed by modifying the model or by adapting the solution strategy.

Modifying the model may include reducing the number of bilinear terms by using transformation techniques that were discussed in section 2.3.4. Reducing the problem size by thoroughly inspecting the model for reducible constraints and variables can lessen computational time. Introducing variable bounds may help decrease the search space.

In this work, it was observed that exploring multiple water minimization techniques simultaneously led to complex mathematical models, even though opportunities can sometimes be found by one technique and not the other. For this reason, there might be merit in exploring water minimization techniques one after another in series, as opposed to exploring them simultaneously.

Adapting the solution strategy may include providing a better starting point for the MINLP problem by using a solution from the relaxed model, RMINLP, can aid with the convergence of the MINLP model. Furthermore, using hybrid solution techniques may also prove to be beneficial. An example of a hybrid solution technique includes that of Dakwala et al. (2014) who presented a combined graphical and mathematical optimization technique to simultaneously optimize a water network along with the energy requirement. In their work, the graphical technique was used to determine values that were then used as parameters for the mathematical program. Hybrid solution techniques can also be simulation-optimization (Sim-Opt), where a simulation model is used to describe the system complexity Lau and Srinivasan (2016).

Due to the advancement in technology, parallel computing where multiple calculations can be carried out simultaneously and web-based optimization platforms can be explored. The computational power of this technology can result in reduced computational times. The free Internet-based NEOS server has more than 60 solvers for numerical optimization provides high-performance parallel computing services, hosted by the Wisconsin Institute for Discovery at the University of Wisconsin, is an example of advanced technologies that can reduce computational times for optimization problems (Czyzyk et al., 1998).

5.5. Possible water saving opportunities

In this work, a central water storage tank is used to store water so it can be indirectly reused or recycled as illustrated by the superstructure in Figure 3.2. Water from different washing operations is allowed to mix in the central water storage tank. As a result of mixing, the overall contamination of water may increase leading to a decline in the number of indirect water reuse or recycle opportunities. Having storage tanks dedicated to specific washing operations such that the wastewater can be indirectly reused or recycled without mixing with wastewater generated from washing other processing units may be worth exploring. This option can, however, prove to be expensive in the short term.

The superstructure in Figure 3.2 also shows that the central storage tank can only receive water from the cleaning in place washing operation and only discharge water to other cleaning operations. The superstructure in Figure 5.1 allows freshwater to be sent to the central storage tank. This can be done to decrease the contaminant concentration in the wastewater, increasing more opportunities for indirect water reuse and recycle. The storage central water tank in Figure 5.1 can also be used to dilute the wastewater to decrease the contaminant levels before disposing it to the environment.

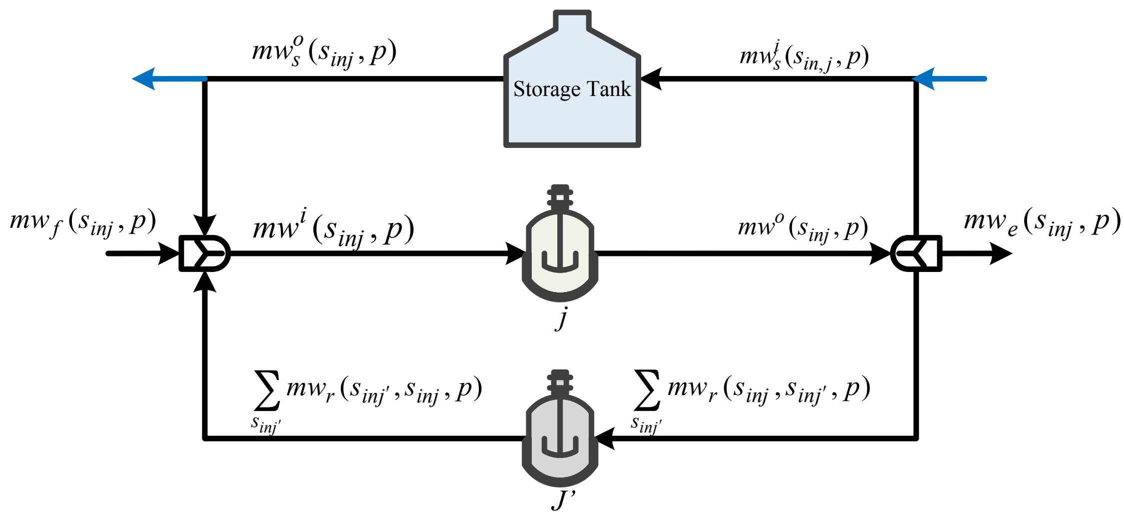


Figure 5.1 Suggested superstructure for indirect water reuse and recycle

One of the conditions that must be met for reuse or recycle to be allowed is that the contaminant concentration from the wastewater producing unit must be less than that of the water receiving unit. It is, therefore, safe to assume that more reuse or recycle opportunities will be available if a regeneration unit that can treat the wastewater before it can be reused or recycled is incorporated. A formulation that simultaneously explores sequence dependent water saving opportunities simultaneously with other wastewater minimization techniques such as regeneration reuse or recycle is therefore recommended. The hypothesis is that greater water savings will be achieved if reuse or recycle opportunities are increased.

Adekola and Majozi (2017) presented a formulation that explored water saving opportunities using sequence dependent changeover times. A central water storage tank or multiple storage tanks, and/or a regeneration unit can be incorporated in their formulation so that it explores direct, indirect, and regeneration reuse and recycle opportunities. Buabeng-Baidoo et al (2017) achieved 85% reduction of wastewater generation by exploring multiple water reuse opportunities, including regeneration reuse by means of a reverse osmosis membrane, in the cleaning in place process of a large scale milk continuous processing plant.

5.6. References

Adekola, O., Majozi, T., 2017. Wastewater minimization in batch plants with sequence dependent changeover. Comput. Chem. Eng. 97, 85–103. doi:10.1016/j.compchemeng.2016.11.016

Buabeng-Baidoo, Mafukidze N., Pal J., Tiwari S., Srinivasan B., Majozi T., Srinivasan R., 2017, Study of water reuse opportunities in a large-scale milk processing plant through process integration, Chemical Engineering Research and Design, 121, pp 81–91, 2017.

Czyzyk, J., Mesnier, M.P., Moré, J.J., 1998. The NEOS Server. IEEE J. Comput. Sci. Eng. 5, 68–75.

Dakwala, M., Mohanty, B., Bhargava, R., 2014. Simultaneous water and energy conservation through graphical and mathematical programming : a case study for fl oat glass industry. J. Clean. Prod. 78, 15–34. doi:10.1016/j.jclepro.2014.04.059

Lau MC, Srinivasan R., 2016, A Hybrid CPU-Graphics Processing Unit (GPU) Approach for Computationally Efficient Simulation-Optimization, Computers and Chemical Engineering, 87, pp49-62.

Majozi, T., Gouws, J.F., 2009. A mathematical optimisation approach for wastewater minimisation in multipurpose batch plants: Multiple contaminants. Comput. Chem. Eng. 33, 1826–1840. doi:10.1016/j.compchemeng.2009.06.008

Savelski, M., Bagajewicz, M., 2003. On the necessary conditions of optimality of water utilization systems in process plants with multiple contaminants On the necessary conditions of optimality of water utilization systems in process plants with multiple contaminants. Chem. Eng. Sci. 58, 5349–5362. doi:10.1016/j.ces.2003.09.004

Chapter 6

CONCLUSIONS

Wastewater treatment options tend to be very expensive hence it is desired to explore wastewater minimization opportunities in production facilities. As an attempt to achieve wastewater minimization, most process integration formulations presented in literature explores the direct or indirect water reuse or recycle without considering the sequence of tasks when determining the amount of water required for washing operations. In this work, the concept of sequence dependent changeover is explored as a wastewater minimization opportunity in multipurpose batch processes. The developed variable schedule continuous-time formulations are unit-specific slot based. A process task and a corresponding washing operation occur in one active time slot. Four scenarios were explored:

- Fixed water requirement with sequence dependent changeover constraints,
- Fixed outlet concentration with sequence dependent changeover constraints,
- Fixed water requirement with sequence dependent changeover constraints and direct or indirect water reuse or recycle, and
- Fixed outlet concentration with sequence dependent changeover constraints and direct or indirect water reuse or recycle.

To validate and demonstrate the applicability of the developed formulations, two illustrative examples with multipurpose batch processes were used. The resultant optimization problems were mixed integer nonlinear program (MINLP) and were solved using a branch and reduce optimization navigator (BARON) solver on the general algebraic mathematical systems (GAMS) platform. A desktop computer with the following specifications was used: Windows 7 Professional, Intel(R) Core ™ i7.4770 CPU @ 3.40GHz, 8.00 GB RAM, and 64-bit Operating System.

All scenarios were applied to illustrative examples and results were compared against the base case. It was found that mathematical formulations that simultaneously explore multiple process integration techniques have a higher chance of achieving significant water savings than those that explore a single technique. For example, there are two formulations that explored sequence dependent changeover opportunities for water minimization simultaneously with direct and indirect water reuse and recycle opportunities that achieved 65% and 61% in water savings respectively.