IMPROVING PROCESS EFFICIENCY THROUGH APPLIED OPTIMIZATION

IMPROVING PROCESS EFFICIENCY THROUGH APPLIED PROCESS SCHEDULING AND PRODUCTION PLANNING OPTIMIZATION

by

Matthew John Hazaras, B.Sc (Eng)

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Applied Science

McMaster University

MASTER OF APPLIED SCIENCE (2011)	McMaster University
(Chemical Engineering)	Hamilton, Ontario, Canada

TITLE:	Improving Process Efficiency Through Applied Process Scheduling and Production Planning Optimization
AUTHOR:	Matthew John Hazaras, B.Sc(Eng) (McMaster University, Canada)
SUPERVISOR:	Dr. Christopher L.E. Swartz and Dr. Tom E. Marlin
NUMBER OF PAGES:	xiv, 101

ABSTRACT

The industrial application of production planning and process scheduling optimization is addressed in this thesis. The first part of the thesis addresses the research into process scheduling application. Several scheduling models are developed based on both discrete and continuous time modelling frameworks. Extensions to both frameworks are presented to address unique production policies and maintenance activities. The potential benefits of schedule optimization is determined through several comparative industrial case studies. The weekly production schedules of the actual plant are compared against the schedules generated by optimization. The historical plant performance is ascertained and areas where efficiency gains are possible are highlighted. In addition, the scheduling model is used to investigate potential changes to production policies.

The second part of the thesis addresses the research conducted in production planning application. The main goal of production planning is the efficient generation of a plan that specifies production targets for products over a medium term horizon. Direct application of previously proposed planning models fails to model several unique and key processing features of the production facility. A production planning model is presented that relaxes the detailed scheduling model structure and exploits the use of traveling salesman type constraints to accurately model sequence dependent changeovers. Two case studies are presented to investigate the benefits of optimization in production planning. The first case study investigates the lowest cost planning solution over a three month planning horizon. The second case study investigates the effects of a key production parameter on the optimality of solution. The results highlight the potential benefit of optimization application in increasing plant processing efficiency and reducing unnecessary production downtime.

Finally, a modelling framework is presented that allows for the combined scheduling of production and maintenance. The framework allows for maintenance with various timing requirements and extends the capabilities of current frameworks.

ACKNOWLEDGEMENTS

I wish to express my gratitude to my supervisors Dr. Christopher L.E. Swartz and Dr. Thomas E. Marlin, who's dedicated support, direction and creative discussions shaped and drove the research project. I also want to acknowledge Dr. Benoit Chachuat who taught me the broad fundamentals of optimization and the potential for optimization to aid all levels of engineering. To Dr. Andrew N. Hrymak, for the early career advice that inevitably lead me to consider graduate studies. I wish to acknowledge the McMaster Advanced Control Consortium and the Department of Chemical Engineering at McMaster University for financial support. Also to Lynn, Nanci and Kathy for all the help in departmental matters.

I would also like to express my gratitude for the Pepsi Company of Canada, specifically Diana Richards, Pat Walsh, Crystal Weese, Stacey Reid, Rhoda Baker and everyone one who gave their time to support the research conducted.

I dedicate this thesis to my family who's support and interest in my education made my achievements possible and built the work ethic which I have today. Also to my love Meredith, who helped me decide to pursue a masters degree and pushes me everyday to be the best I can be.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Consumer Packaged Goods (CPG) industry is made of products with high turnover rates and low to moderate prices. These products are typically sold in grocery stores, membership stores and occasionally to food service providers. The CPG market sector is also categorized by high profit margins, low barriers to entry, and a highly competitive market place. This competition provides incentive for efficient process operation and production planning, as it provides means to lower the company bottom line and improve profitability. This incentive is reflected through increased collaboration between industry and academia to explore new techniques and methods to improve process efficiency.

1.1 Motivation and Goals

This research is a collaborative effort with PepsiCo Canada to highlight and improve operational performance in a CPG production facility. The first goal is to investigate the benefits of scheduling optimization on an industrial level. Production scheduling is conducted on a short-term horizon spanning one week. Optimization models based on both discrete and continuous time formulations are presented. Historical information is used to develop four case studies and benchmark plant performance, where the objective is to minimize the production makespan. The potential improvements in operational efficiencies are quantified through comparison to historical schedules.

In addition to providing solutions to short-term process scheduling, the developed optimization models can be used to evaluate potential future plant operations. The investigation of two operational policies after plant retrofit is conducted through modification of scheduling optimization models. The improvement in plant throughput is investigated for each operational strategy and plant efficiency gains highlighted.

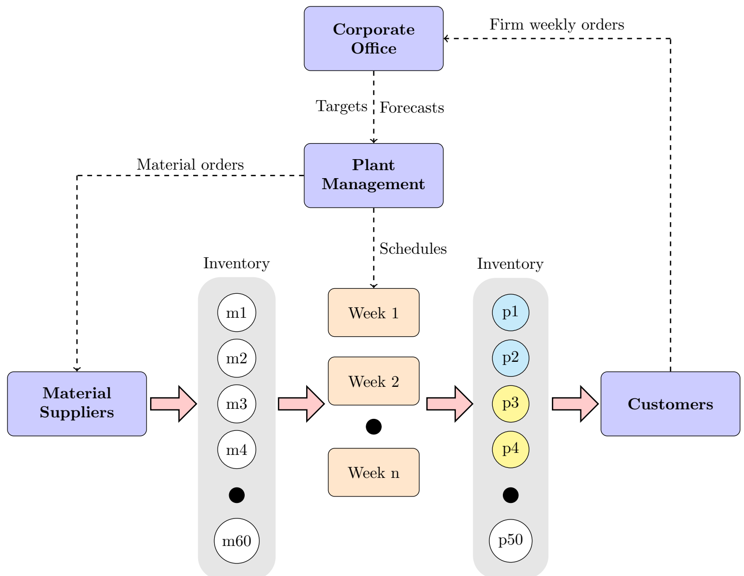
The second goal of this research is to move from short-term scheduling to medium-term production planning optimization. Due to the increase in the time horizon, production planning models are typically modelled using less detail than scheduling models. This can pose a significant problem of capacity overestimation if critical plant characteristics are modelled inaccurately. A production planning model is developed which exploits traveling salesman based constraints to model sequence dependent changeovers within the planning intervals. The planning model developed accurately reflects the scheduling level detail while providing a method to investigate longer term horizons. Investigation of the broader supply chain is conducted through industrial case studies developed from historical demand and inventory data. Efficiency gains are highlighted and possible improvements to the production planning process are discussed.

1.2 Industrial Setting

This research focuses on a production facility manufacturing a consumer food product. The supply chain in which the plant is a part of is shown in Figure 1.1, while the production facility layout is illustrated in Figure 1.2. The product supply chain consists of the production facility with customer demand and material supplier nodes. The corporate office receives weekly orders from customers and forecasts future product demands. Production targets and demand forecasts are then sent to plant management who utilize the data and generate applicable weekly production schedules and place material orders. Inventory of both raw materials and product is kept on site to satisfy customer demands.

The facility produces on the order of 50 different stock keeping units (SKUs) which are based on roughly 30 product material formulations. Each material formulation is associated with a group of product SKUs which are varied in packaging size; this group is considered a

Figure 1.1: Product supply chain layout



product family. Similarities between product families exist in the product formulations, as key ingredients lead to similar allergen concerns and cross-product contamination issues. Due to these similarities, product families are segregated into 6 groups, herein referenced as *production groups*. If production is to change between these groups, a plant wide cleaning operation must be conducted for quality control reasons.

The production facility consists of four main areas: raw material storage, a mixing area, a packaging area and product storage facilities. The overall production process is depicted in

Figure 1.2: Production facility layout

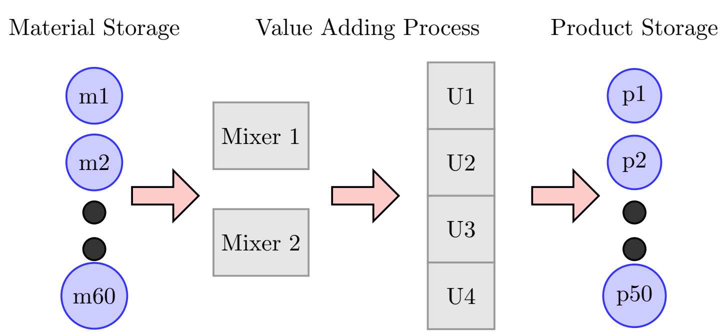


Figure 1.2. The mixing train receives from storage raw materials to be portioned and mixed. The mixed intermediate material is then transferred to the packaging area for finishing. The packaging area consists of four packaging lines; three packaging lines (U1, U2, U3) are reserved for the packaged products (small, medium, large). The fourth packaging line U4 is used to finish specialized products. Physical limitations allow only one packaging line to be connected to the mixing train at a time, with the restriction of no intermediate storage; thus the packaging lines and mixing train effectively act as a single unit. The finished products are sent to storage for holding before shipment to meet customer demands. Changeovers in processing operations are sequence dependent and can be classified, by clean duration, into three types: a 15 minute, 1 hour, and 3 hour clean. The 3 hour clean is a plant wide cleaning operation where all processing equipment are taken offline for sanitation. It is required that this cleaning operation be conducted at least every 36 hours regardless of production sequence.

Several unique production policies also restrict the manner in which production is scheduled. Such policies include: i) all products within a product family are grouped together in the scheduling horizon, ii) product families with similar material compositions are run together in production groups, which are separated by plant wide cleaning operations, and iii) production tasks should not be duplicated within a production schedule.

1.3 Main Contributions

1. **Scheduling Optimization Modelling** Extensions to both discrete and continuous time scheduling optimization models are presented. General extensions to include production and product family grouping are presented for both the discrete and continuous time models. Extension of the continuous time model to incorporate process cleaning is presented and is applicable for use as a general combined maintenance and scheduling model.

2. **Historical Plant Benchmarking** Through comparisons of actual plant schedules to optimized results, this thesis effectively benchmarks the historical performance of the processing facility. Through this methodology the benefits of scheduling optimization to process scheduling become even more apparent.

3. **Production Planning Modelling** A general production planning model used to account for sequence dependent changeovers is modified in a manner similar to the aggregate scheduling model proposed. Extensions allow for accurate production planning to be conducted for a 12 week planning horizon. Case studies examining the effectiveness of the production planning methodology of the processing facility are presented and conclusions drawn.

1.4 Thesis Overview

Chapter 2 – Literature Review

This chapter provides an in-depth discussion on the aspects of process scheduling, combined maintenance and scheduling optimization, and production planning optimization.

Chapter 3 – Process Scheduling Model Development

The main focus of this chapter is on the development of scheduling optimization models. Both discrete and continuous time models are developed and extensions to published scheduling optimization models are presented. One of the models developed exploits model reformulations through aggregation of scheduling tasks into explicit product families. This aggregation allows for a reduction in model size and is intended to reduce the number of inherent symmetrical solutions in the general continuous time optimization model.

Chapter 4 – Scheduling Case Studies

This chapter presents the application of the developed scheduling models to the industrial facility through evaluation of four case studies. The case studies are developed using historical processing and demand information. The results of the optimization are compared against the actual historical schedules generated by plant personnel. The efficiency gains possible are highlighted and historical performance is benchmarked against the optimal process schedules.

During the term of this research the target production facility was to undergo retrofitting to improve production capacity. Two future production strategies designed to exploit this new capacity were proposed by plant management. This chapter investigates the application of the aggregate continuous time scheduling model for evaluation of both production strategies. The aggregate model is augmented with additional constraints reflecting the policies of each strategy, and results are compared to determine the benefits and drawbacks of each strategy. Through this use of optimization models the best case production scenarios of each operational strategy can be evaluated and provide valuable information for plant management.

Chapter 5 – Production Planning

The efficiency gains possible at the short-term scheduling level provide incentive to investigate the longer-term planning level. This chapter presents the development of a production planning model that accurately approximates the detailed scheduling formulation. The planning model is used to investigate two industrial based case studies. The first case study investigates the application of the planning model to a three month horizon and contrasts the profitability of the facility with and without specified production targets for key materials. The resultant production plans are summarized through the weekly metrics used by the production facility and possible process improvements are discussed. The second case study investigates the effects of varying a key operational parameter that limits the length of time before a process wide clean must be conducted.

Chapter 6 – Chapter Conclusions and Recommendations

Concluding remarks on the applicability of optimization for use in production planning and scheduling are given. Results are reiterated and main remarks drawn, with specific benefits to the processing facility addressed. Possible avenues of future research are highlighted.

Chapter 2

Literature Review

The intent of this chapter is to provide a review of the important concepts involved in this research. Three areas are explored: process scheduling, combined maintenance and process scheduling, and production planning. In each case the reader will be provided with a background of each topic and an overview of previously completed research.

2.1 Process Scheduling

Scheduling of batch and semi-continuous processes is tied intimately with the efficiency of processing operations in industrial manufacturing plants. Critical performance measures, such as equipment utilization and customer demand satisfaction, are influenced directly by the process schedule. The complexity of many multi-purpose or multi-product manufacturing plants makes scheduling a non-intuitive task, complicated by such things as: equipment networks, complex product recipes, storage limitations, product due dates (material delivery) and utility restrictions. Given the above considerations, the development of mathematical models to optimally schedule processes has received considerable attention both from industry and the process systems engineering community.

The application of scheduling optimization models to improve the efficiency of an industrial processing plant is a central focus point in this thesis work. The rest of the chapter is organized as follows. First, the nature of scheduling problems is reviewed, followed by a discussion of the main modelling challenges associated with schedule model feasibility and optimality. Then, the two main scheduling frameworks adopted later in the thesis are reviewed. Concluding remarks on the benefits and drawbacks of such modelling methodologies with respect to the application to industrial problems, are addressed last.

Classification of Scheduling Problems

Many issues need to be addressed when developing optimization models for the scheduling of batch and semi-continuous processes. The dynamics and policies of any manufacturing plant must be accurately represented mathematically if the results generated are to represent optimal (feasible) process schedules. Méndez *et al.* [2006] highlighted 13 major categories in which the dynamics of common manufacturing plants fall under. These categories include process topology, equipment assignment, equipment connectivity, material storage policies, batch sizing, processing time, changeovers, resource constraints, and the degree of certainty. Select categories are discussed below in further detail.

The equipment topology and connectivity of a processing plant can significantly affect the complexity of scheduling optimization models. Manufacturing plants are typically categorized as single-stage, multi-stage or sequential processes and consist of one or many parallel units at each stage. Single-stage and sequential processes follow a defined step-by-step production pathway common for all products. This processing structure can be exploited for model reduction as the step-by-step nature admits for simplified model development. In multi-stage processes, product pathways are not necessarily common and may utilize different process equipment. Processing complexities such as batch splitting, product mixing and recycle loops may exist. Such complexities often require a greater level of detail to be included within scheduling optimization models. Multi-stage/multi-purpose modelling methodologies can be used to represent single-stage and sequential processes.

It is also necessary to ensure that the policies that dictate the storage of materials between equipment units are also upheld. In many industries these material transfer policies can be classified into several of the self-explanatory categories, as listed below. Failure to account for proper policies will lead to infeasible schedules and the possible production of off-specification products.

- unlimited intermediate storage (UIS)
- no intermediate storage (NIS)
- finite intermediate storage (FIS)
- zero wait (ZW)

In addition to the above plant dynamics the objective of the optimization problem must be considered. Common objectives include, makespan minimization (completion time of the last production task), profit maximization, job tardiness minimization and production maximization.

Classification of Scheduling Models

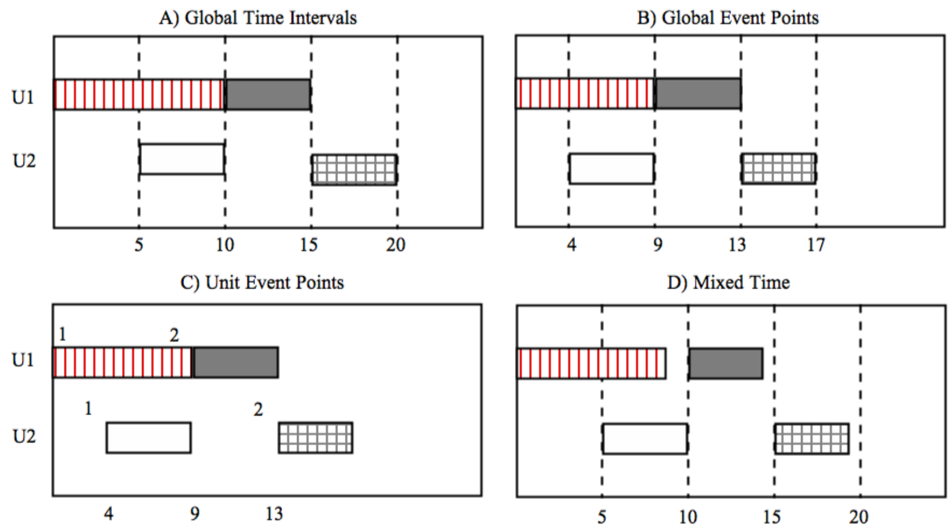
The many characteristics of process scheduling problems has led to the development of multiple modelling frameworks, some capable of representing generic problems, other designed on a situational basis. Multiple equivalent ways of modelling the same problem exist, and particular models will be computationally more efficient than others on a problem to problem basis. The key differences in modelling methodologies lie in the representation of time, processing events, and the handling of material balances. Understanding the key differences in modelling methodologies, what problems they can be applied to and the expected benefits (drawbacks) will help form a solid foundation of modelling knowledge in scheduling optimization.

Representation of Time and Processing Events

The main differentiation between modelling methodologies for short-term scheduling lies in the representation of time. Scheduling optimization can be modelled in discrete, continuous or mixed time representations. Each methodology has advantages and drawbacks when compared to one another; the rest of the section will briefly introduce each methodology. The reader is referred to the review of Floudas and Lin [2004] to cover the topics of time representation in further detail.

Discrete time models subdivide the scheduling horizon into evenly spaced intervals and allow events to occur only at interval boundaries. A key contribution to discrete time scheduling

Figure 2.1: Different scheduling model methodologies for the representation of time and processing events



models was made by Kondili *et al.* [1993] with the introduction of the discrete time state-task network (STN) paradigm. An example of a discrete time schedule is depicted in Figure 2.1 A). The production tasks (shaded rectangles) can only begin and end at the boundaries of the time intervals. The discretization of the time horizon allows for constraints to be monitored at predefined intervals, a simplification that reduces problem complexity and allows for simple model structures. Discrete time models are considered data dependent as interval length is commonly chosen as the duration of the shortest processing task.

Continuous time models were developed to remove such data dependancy and reduce the number of binary and continuous variables required to model a process schedule. The exact timing of events is introduced as an optimization variable and is allowed to vary throughout the horizon. Multiple modelling methodologies utilizing a continuous time representation have been developed over the years. An advantage of continuous time modelling allows for the timing and duration of tasks to be variable, this allows for improved modelling of semi-continuous type plants. Relevant to this work are the global event point and unit event point modelling representations, presented by Maravelias and Grossmann [2003b] and Ierapetritou and Floudas [1998], respectively.

Global event points is a version of global time intervals where the location of event points, common and shared by all unit resources, is allowed to vary throughout the horizon. Figure 2.1 B) depicts a global event point schedule, where tasks can begin and end at any point within the scheduling horizon. The task durations differ in comparison to the discrete time schedule example and the schedule can finish earlier. It is noted that one event point is needed to model the start and finish of tasks; in this example 5 event points are needed to model the schedule.

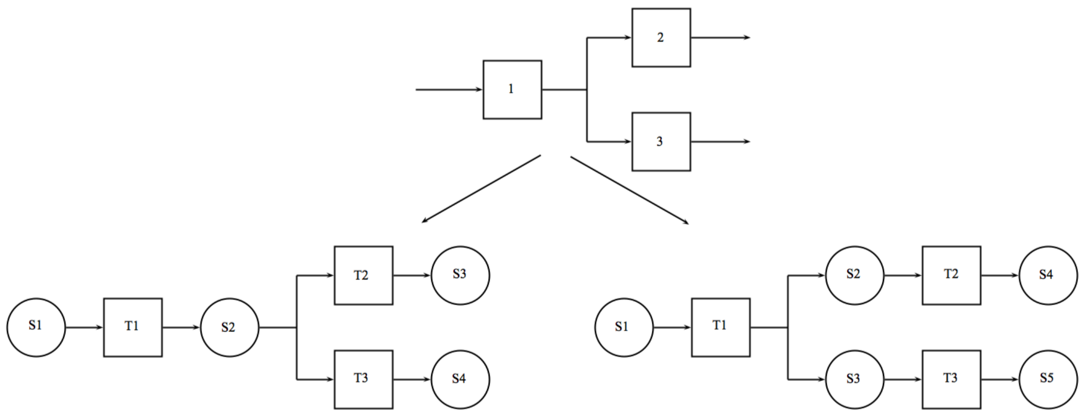
Unit event points specifies a set of heterogenous points for each unit, where the timing of an event point is allowed to vary from one unit to another. This behaviour can be seen in Figure 2.1 C), event point 1 occurs at time 0 on unit U1 and at 4 hours on unit U2. This methodology allows for a reduction in the number of event points needed to model a schedule. In this example only two event points are needed, as compared to the 5 needed with the global event point methodology.

Maravelias [2005] developed a mixed time modelling methodology to address the common limitations of both discrete and continuous time models. In mixed time models the horizon is sub-divided into equal intervals but production tasks have variable processing times and span an unknown number of time periods. In this manner semi-continuous plants can be represented more accurately than with pure discrete time models. The proposed model handles back-order and inventory holding costs linearly, which is not possible in continuous time models. In addition due dates and material delivery are modelled with no additional computational cost. Figure 2.1 D) depicts a mixed time schedule; the time intervals are fixed but tasks can span multiple intervals. The processing tasks are allowed to finish at or before the end of the last processing interval. By not assuming that tasks occupy the entire interval allows for better batch-size representation. The model still exhibits data dependency as optimality can be effected by the choice of interval length.

Material Balances

Many modelling frameworks in the process systems engineering literature are developed to allow simultaneous optimization of both the number of and size of batches used within the scheduling horizon. These monolithic approaches are capable of handling generic processing structures and such tasks as batch mixing, splitting and recycle loops. The state-task network (STN) is a comprehensive scheduling framework introduced by Kondili *et al.* [1993].

Figure 2.2: Process-flow-diagram converted into two state-task networks, adapted from Davies [2008].



The state-task network is a representation of a process flow diagram, where material states (circles) and processing tasks (squares) are shown as a connected graph. The state-task network removes the ambiguities of a process flow diagram and allows for the reader to easily follow the flow and transition of materials through the processing steps.

An example of the ambiguity in a process flow diagram is depicted in Figure 2.2. The process flow diagram shows the location and topology of the process equipment, but does not show the materials or intermediates being processed. Two possible state-task networks are derived from the process flow diagram. On the left, one material is produced after T1 which is split and operated on by T2 and T3 to produce two different product states. On the right, two materials are produced after task 1 (T1) and sent to separate processing tasks (T2 and T3) for finishing. As is seen the state-task network removes this unintentional ambiguity and allows readers to easily follow material transitions.

Comparative Analysis of Models

There is a vast amount of literature on the topic of optimal short-term scheduling of batch and semi-continuous processes. The applicability of various modelling methodologies lies in the dynamics the model is capable of representing and on the events that occur during the scheduling horizon. However, multiple modelling methodologies may still be applicable to any particular problem or processing facility. The choice of modelling methodology then becomes a choice on computational efficiency, for which certain models will outperform others.

Discrete Time Models The efficiency of discrete time models depends largely on the data of the problem at hand. The number of time intervals required is a function of the scheduling horizon length and time interval length. Small interval durations allows for additional flexibility in the timing of events, but increases the model size (binary variables and constraints). However, it has been shown by Maravelias and Papalamprou [2009] that in certain cases the choice of finer discretization, albeit increasing model size, leads to improved computational performance.

Another drawback to the discrete time modelling methodology is the drop in performance when sequence dependent cleaning tasks are required. The number of constraints necessary to enforce sequence dependent cleaning tasks is related to the number of tasks and the number of time intervals. Therefore, if a scheduling model exhibits a large number of processing tasks or has a fine time discretization the model sizes may quickly become computationally intractable. In addition, tasks with batch size dependent processing times cannot easily be accounted for in discrete time models. This limitation makes modelling semi-continuous processes difficult.

Continuous Time Models Global event based and unit event based continuous time models are considered the two main time representations for general network based short-term scheduling problems. The computational efficiency of the global event based and unit event based models is dependent on the dynamics that need to be considered within the model. Global event based models are considered to have the most general and most rigorous representation of time [Janak *et al.*, 2004]. However, the heterogeneous nature of the unit event based models will require fewer event points to model a schedule. This reduction in event points reduces model sizes and results in better computational performance. Unit event based models also provide a simple and effective manner to address sequence dependent cleaning requirements within the core model. Global event based models require the modification of the core timing constraints to represent sequence dependent cleaning, and additional event points will be required to model a schedule. This can lead to degraded computational performance as the number of event points is considered as a limiting factor in continuous time models [Méndez *et al.*, 2006].

The heterogeneous nature of the unit event based models may also require the specification of additional storage tasks to accurately represent some plant layouts [Maravelias and Grossmann, 2003b]. The non-common time grid of the unit event based models also complicates the ability to consider resources constraints, such as heating or cooling utilities, or product due dates within the interior of the time horizon. An extended formulation of the unit event based models was present by Janak *et al.* [2004] to address the above drawbacks of the unit event based models. Maravelias and Grossmann [2003a] proposed an extension to the novel global event based model such that material receipt and product delivery dates can be effectively modelled. Material receipt has yet to be addressed within the unit event based models.

Shaik *et al.* [2006] present a comparative analysis of multiple continuous time models, including both global event point and unit event point models. In general, unit event based models outperform global event based models when resource limitations are not considered; and if sequence dependent cleaning is required. Unit event based models have also been shown to outperform global event based models when resource limitations are considered. However, the required additional variables and constraints reduces the computational performance gap between the models. Unit event based models with due date considerations require an additional event point for each order due, whereas global event based models can allow multiple due dates to occur at the same event point. Unit event based models have not been extended to address the receipt of materials within the scheduling horizon, an aspect that global event based models can effectively model.

Mixed Time Models The mixed time model of Maravelias [2005] is capable of handling resource limitations, sequence dependent cleaning, product due dates and material receipts. In addition, mixed time models can handle backorder and inventory costs linearly, whereas continuous time models can not. However, the discretization of the time horizon has the potential to lead to sub-optimality; large discretization may lead to poor unit utilization and unnecessary down time. Conversely, fine discretization leads to model sizes that quickly become intractable.

2.2 Combined Maintenance and Process Scheduling

In industry, reliable plant operation can make the difference between fulfilling customer orders and incurring contractual penalties. A key aspect of reliable operation is equipment performance. To keep equipment units running at satisfactory performance levels, preventative maintenance is often carried out to avoid untimely breakdowns. Often manufacturing plants operate continuously, and maintenance must be preformed in conjunction with production tasks. These maintenance activities impose restrictions on equipment units, reducing available processing time. In multipurpose batch manufacturing facilities, equipment downtime can affect the production of multiple product pathways and have a significant effect on plant profitability. Thus the integration of maintenance and production scheduling is expected to yield improved production schedules with maintenance that has reduced impact on plant performance.

Several developments on the integrated maintenance and scheduling problem are presented by the operations research community. Lee [1996] studied the problem of production scheduling on single and parallel machines with maintenance events of fixed timing. The formulation is capable of both resumable and non-resumable production jobs, but can handle only one maintenance event. Liao and Chen [2003] addressed the problem of scheduling production with multiple maintenance events of fixed timing. The formulation is based on a single machine framework with non-resumable jobs. Qi *et al.* [1999] address the issue of flexible maintenance scheduling on a single equipment unit. The authors specify that the equipment unit has a maximum allowable continuous processing time and that maintenance must be scheduled before this limit occurs. The number of maintenance events is an optimization variable and processing jobs are non-resumable. Chen [2008] addresses the issue of flexible maintenance events where the number of maintenance events is known. The events are allowed to occur within a specified time window and processing jobs are non-resumable.

The above formulations all deal with single or parallel machine facilities, focusing on the scheduling of specific dedicated production tasks. An industrial problem of greater relevance to the process industry is the scheduling of batch manufacturing facilities with multipurpose equipment in complex networks. Such configurations are common in the specialty chemical, pharmaceutical and consumer goods industries. A key contribution to the optimization of such plant configurations was made by Kondili *et al.* [1993] with the introduction of the state-task network (STN) paradigm. The formulation relies on a discrete representation of time, dividing the scheduling horizon into a known number of equal length intervals. Production tasks are allowed to begin and end only at the boundaries of the specified intervals. Multiple constraint sets are proposed to handle task-unit allocation, storage limitations, processing utility restrictions, batch size restrictions, product due dates and material receipts. The formulation also addresses temporary unavailability of equipment units to account for maintenance events of fixed timing.

Dedopoulos and Shah [1995] present a discrete-time STN based scheduling framework that addresses the combined optimization of production and maintenance tasks. The framework also considers available maintenance crews and their associated costs. As an example, the authors optimize the maintenance and production scheduling of a lubricant manufacturing facility. Davies [2008] also addresses the issue of scheduling flexible maintenance in a multipurpose batch plant. The author's formulation is based on a discrete-time STN paradigm and specifies that a known number of maintenance events must occur within the scheduling horizon. Maintenance events are allowed to occur within a window of time intervals and are scheduled in conjunction with production processing. The formulation considers various objective functions, including makespan minimization and throughput maximization, and investigates the benefits of flexible versus fixed maintenance events.

Representing time as discrete intervals is an approximation by definition and can lead to sub-optimal solutions [Floudas and Lin, 2004]. In addition, discrete-time models are known to have difficulty solving problems with sequence-dependent changeovers and variable batch processing times, two features common in industry. Continuous-time modeling frameworks were developed to address the limitations of discrete-time models and introduce the timing of processing events as optimization variables. Continuous-time formulations are also divided into sub-categories based on the representation of processing events and can be classified as slot, global event and unit event based models. Two key contributions to continuous-time scheduling of multipurpose batch plants are the unit event based model of

Ierapetritou and Floudas [1998] and the global event based model of Maravelias and Grossmann [2003b]. Although the above formulations provide different methods to address the general multipurpose batch scheduling problem, neither formulation addresses the inclusion of maintenance events. Mockus and Reklaitis [1999] proposed a continuous-time STN modeling formulation for scheduling of multipurpose batch and continuous plants. Maintenance events with fixed timing are included as planned down-time on equipment units. The formulation results in a MINLP which poses computational challenges for large problems.

2.3 Production Planning

Production planning is concerned with the optimization of high-level decisions, such as facility production targets, raw material orders and product inventory levels, such that manufacturing capital is utilized efficiently. Production planning is also called tactical level optimization in most supply chain literature and is concerned with time horizons of several months to a year. Many manufacturing facilities operate in a batch or semi-continuous fashion on multi-purpose equipment, which allows for multiple products to be produced within a single facility. This additional manufacturing flexibility allows for production to be changed to meet fluctuating customer demands. As a significant number of decisions need to be made over the medium-term horizons, multi-purpose production systems lend themselves to the application of optimization techniques. It has been of interest to the process systems engineering community to develop mathematical models capable optimally planning production of multi-purpose/product plants over extended time horizons.

Early work on the planning of multi-product plants was presented by McDonald and Karimi [1997], where a MILP model is presented to optimally plan multi-site production supply chains based on the classic economic lot-sizing planning problem. Extensions are presented to account for minimum production run-lengths, customer-plant product sourcing decisions, time lag between supply chain nodes and grouping of products into production families. Several examples based on industrial data are presented for illustrative purposes. Further developments to the general capacitated lot-sizing model is presented by Sung and Maravelias [2008]. The formulation includes a novel time-bucket representation to account for uniform or non-uniform interval lengths, production of multiple products in one interval and product set-ups that can span time intervals. The model assumes sequence independent changeovers and should be applied to single-stage like processing operations.

An issue with using the solutions from planning level models directly in practice the is possibility of overestimating production capacity. A method to avoid this problem is the integration of planning and scheduling level models. Verderame and Floudas [2008] present a production planing model that determines the daily and period production targets for multi-purpose, multi-product batch plants. The model is described as a planning with production disaggregation Model (PPDM). The authors integrate the model with a unit-specific continuous time scheduling model in a rolling horizon manner to facilitate the medium-term planning and scheduling for a production facility. The authors further extend the formulation to consider multi-site production networks, as presented in Verderame and Floudas [2009]. Another method to integrate the planning and scheduling levels is the development of surrogate scheduling models that are included in the planning level. Sung and Maravelias [2007] proposed a projection based method to find the convex hull of feasible production targets and an over-estimation of production costs. The feasibility region of production targets, the production attainable region (PAR), is incorporated into the planning levels as additional constraints and provides all the necessary information to accurately describe the capacity of the production plant. Although the above method provides tight bounds on the feasible region of production targets the method fails to account of the non-convexitiy of many scheduling problems. Sung and Maravelias [2009] extended the projection based method to incorporate non-convexities into the feasibility space. Two MILP formulations capable of representing the non-convex feasibility region of the scheduling model are presented. Through example comparisons it is shown that the use of non-convex feasibility regions improves the accuracy of the surrogate models and allows for additional efficiency gains when solving planning levels problems.

The true capacity of such plants can be hard to determine in practice as it can depend on many factors, one example of which is plants that exhibit sequence dependent changeovers. In such situations the true capacity of the plant is a function of not only the operating horizon but also the sequence and number of products planned to be produced. Erdirik-Dogan and Grossmann [2007] proposed a novel production planning model that accounts for sequence dependent changeovers explicitly, through the use of traveling salesman (TSP) based constraints. Changeovers are accounted for by generating cyclical production sequences and breaking the cycle at the link with the highest changeover. Although the method accounts for sequence dependent changeovers explicitly a possible drawback is the occurrence of sub-cycles. Although the authors present an iterative method to introduce sub-cycle elimination constraints it has been shown in instances to produce sub-optimal results [Castro *et al.*, 2008]. Further use of traveling salesman based planning models have been presented by Liu *et al.* [2008].

Chapter 3

Scheduling Model Development

This chapter presents the development of three scheduling optimization models capable scheduling operations at the target production facility. The first model developed is based on the discrete time state-task network framework proposed by Kondili *et al.* [1993] and Shah *et al.* [1993]. The second model developed is based on the global event continuous time state-task network framework proposed by Maravelias and Grossmann [2003b]. Finally an aggregation technique is used to simplify the global event continuous time model such that problem knowledge can be exploited to improve computational performance.

3.1 General Discrete Time Model

The global time interval framework divides the scheduling time horizon into equal length intervals, of length Δt, and is indexed as the set $t = \{1, \ldots, H+1\}$. H is a parameter defining the total horizon length, and each time interval is common for all plant resources. States are represented as index s, accounting for all material, intermediate and product states. Production tasks are indexed as i and processing units are indexed as j. Each processing task i is associated with a specified processing time (pt_i) and must begin and finish at an interval boundary. The sets $j \in K_i$ and $i \in I_j$ are introduced as the set of units capable of processing tasks i and the set of tasks which can run on unit j, respectively. Allocation of production tasks to process equipment throughout time is tracked through binary variable W_{ijt}, which is defined as 1 if task i is assigned to start in unit j at the beginning of time interval t.

It is noted that the target facility exhibits several unique production policies that have not been incorporated into the original scheduling methodology of Kondili *et al.* [1993]. These policies include 1) a recurring process wide cleaning operation conducted at most every 36 hours, 2) grouping production tasks within the same family together, and 3) processing similar product families in groups between the recurring process wide cleans. This section will present the discrete time state-task network framework used to model the target facility and present extensions to the model to account for the production policies stated above.

Allocation Constraints

In scheduling formulations it is necessary to ensure the proper unit-task allocation, such that plant resources are properly utilized. The *full backward* allocation constraint of Shah *et al.* [1993] was shown to improve computational performance in comparison to the original constraint proposed in Kondili *et al.* [1993], and as such will be used in the formulation. The allocation constraint is given below:

$$\sum_{i \in I_j} \sum_{t'=t}^{t-pt_i+1} W_{ijt'} \leq 1 \qquad \forall j,\ t \tag{3.1}$$

Constraint (3.1) ensures that only 1 task is active on unit j during time interval t by referencing the all tasks $i \in I_j$ within the *backward* interval of $t - pt_i + 1$ and requiring only one to be active. In this manner if task i' started processing at a previous time interval t' and is not finished at interval t constraint (3.1) would restrict all other binary variables W_{ijt} within $i \in I_j$ for this interval to be zero.

It is important to note the production facility operates in a semi-continuous nature, where the processing time is dependent on the batch size of a task. Although the discrete time formulation can not directly include variable task processing times it is possible to determine process time prior to optimization. The discrete formulation will consider the objective of makespan minimization, and as such will produce as much product as demanded and no more. As such the processing time pt_i is determined as the required batch size of task i

divided by the processing rate.

Unit Capacity Limitations

The batch size of task i will be restricted based on the limitations of equipment unit j. In addition storage capacity is often finite and state inventory must not be allowed to accumulate over this limit. The batch size of task i on unit j at time interval t is defined through variable B_{ijt}. The following set of constraints is proposed:

$$B_{ij}^{min}\, W_{ijt} \leq B_{ijt} \leq B_{ij}^{max}\, W_{ijt} \qquad \forall i,\ j \in K_i,\ t \tag{3.2}$$

Constraint (3.2) enforces the batch size of task i on unit j at time interval t to remain within the minimum and maximum unit limits, as defined by parameters $B_{i,j}^{min}$ and $B_{i,j}^{max}$.

$$C_s^{min} \leq S_{st} \leq C_s^{max} \qquad \forall s,\ t \tag{3.3}$$

Constraint (3.3) limits the inventory of state s at time interval t to be within the minimum and maximum inventory specifications, as defined by parameters C_s^{min} and C_s^{max}.

Material Balance

As batch tasks are executed a relative proportion of material state s will either be consumed as input (ρ_{is}) or produced as a output ($\bar{\rho}_{is}$). This state utilization means the inventory levels of states s will change throughout the scheduling horizon, and as such it is necessary to balance the utilization of states s. The following constraint is imposed:

$$S_{st} = S_{s,t-1} - D_{st} + \sum_{i \in TP_s} \sum_{j \in K_i} \bar{\rho}_{is}\, B_{ij,t-pt_i} - \sum_{i \in TC_s} \sum_{j \in K_i} \rho_{is}\, B_{ijt} \qquad \forall s,\ t \tag{3.4}$$

Constraint (3.4) enforces that the current quantity of state s at time interval t (S_{st}) is equal to the previous quantity of state s ($S_{s,t-1}$) minus product demands satisfied (D_{st}) plus the proportion of output material generated from previously processed batches ($\bar{\rho}_{is}\ B_{ij,t-pt_i}$) minus any input material used for current batches ($\rho_{is}\ B_{ijt}$).

Sequence Dependent Cleaning

Sequence dependent cleaning between production tasks on processing equipment is a common requirement in many industries. If cleaning tasks do not require the use of shared utilities then cleaning tasks can be modelled by ensuring sufficient time is left for a unit to be cleaned between processing tasks. The duration of a changeover from task i to task i' on unit j is represented through parameter $sl_{jii'}$. The follow constraint is imposed:

$$\sum_{i' \in I_j} \sum_{t'=t+pt_i}^{t+pt_i+sl_{jii'}-1} W_{i'jt'} \leq M(1 - W_{ijt}) \qquad \forall j,\ i \in I_j,\ t \tag{3.5}$$

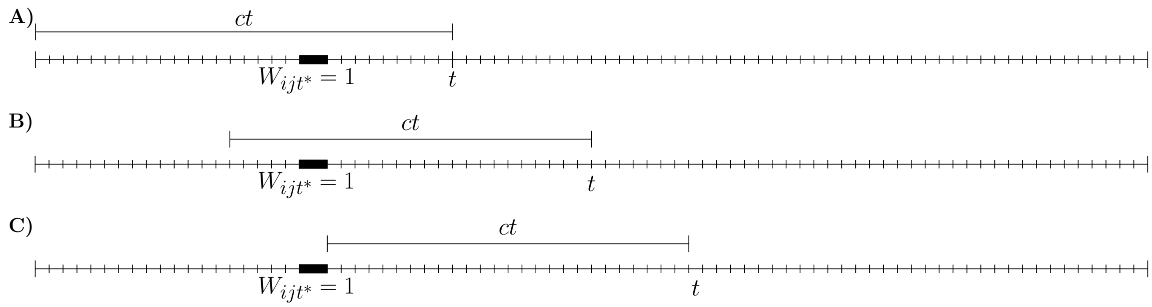
If unit j starts processing task $i \in I_j$ at time t, no other task $i' \in I_j$ can begin for $sl_{jii'}$ time intervals after the end of the first task, and is modelled as constraint (3.5). If parameter $sl_{jii'} = 0$ then the above constraint is a redundant assignment constraint.

3.1.1 Extension to Recurring Plant Wide Cleaning Operations

As stated in Section 1.2 the production operations at the target facility require the process to be taken offline at least every 36 hours for a process wide clean. The cleaning operations can be considered a type of recurring maintenance and is included in the scheduling formulation through the creation of a new task $i \in I$ which has a processing time equivalent to the duration of the process wide cleaning operation. Let I_c represent the set of all such cleaning tasks, as the task is "process wide" the task $i \in I_c$ is included into set I_j for all units j. The process wide cleaning task i must recur at least within $ct = \frac{36}{\Delta t}$ time intervals following the completion of the previous process wide cleaning task. The following constraint is imposed:

$$\sum_{t-ct-pt_i+1}^{t'=t} W_{ijt'} \geq 1 \qquad \forall j,\ i \in I_j \cap I_c,\ t \geq ct \tag{3.6}$$

Figure 3.1: Discrete time recurring cleaning event, A) example of first constraint, B) recurrence of cleaning event before ct time intervals, and C) recurrence of cleaning events at ct time intervals.



Constraint (3.6) enforces cleaning task i to occur within ct time intervals from the start of the horizon and within ct time intervals of the completion of previous cleaning task ($ct + pt_i$ from the start of the previous cleaning task). The first occurrence of constraint (3.6) is defined at time interval ct, and the summation term is enforced from t to $t - ct - pt_i + 1$. This enforces at least one cleaning task to be active between time intervals 1 to ct, as portrayed in Figure 3.1 A). After the first constraint it is required that a cleaning task recur within ct time intervals from the completion of the first task.

It is possible for cleaning task i to recur within the ct time interval limit of the previous occurrence. In Figure 3.1 B) time interval t remains within ct time intervals from completion of the previous cleaning task. Therefore it is not necessary for cleaning task i to recur, but if determined optimal the task may be assigned to begin within this time period. If ct time intervals have passed from the completion of the previous cleaning task ($ct + pt_i$ from the start of the previous task), constraint (3.6) will ensure the cleaning task i recurs, as depicted in Figure 3.1 C).

3.1.2 Extensions to Production Policies

The target facility follows three sets of production policies related to the relative ordering of processing tasks within the production schedule. These are: 1) all tasks within a product family must be run sequentially within the schedule, 2) processing tasks with similar material formulations are run together between plant wide cleans, and 3) no duplication of production tasks is allowed.

Product Family Grouping

It is necessary that production tasks i within the same product family f be processed as a group within the production schedule. To accomplish this, new variables are introduced to track the activity of product family f at time interval t. Fa_{fjt} is defined as 1 if production family f on unit j is active at time interval t, and Ff_{fjt} is defined as 1 if production family f finishes on unit j at time interval t. F_j is introduced as the set of product families f capable of production on unit j. The following constraints are imposed:

$$Fa_{fjt} \geq \sum_{i \in I_f \cap I_j} \sum_{t'=t}^{t-pt_i+1} W_{ijt'} \qquad \forall j,\ f \in F_j,\ t \tag{3.7}$$

Constraint (3.7) enforces product family f on unit j to become active if any task $i \in I_f$ is beginning or actively processing at time interval t.

$$Fa_{fjt} \geq Fa_{fj,t-1} - Ff_{fjt} \qquad \forall j,\ f \in F_j,\ t \tag{3.8}$$

$$Ff_{fjt} \leq Fa_{fj,t-1} \qquad \forall j,\ f \in F_j,\ t \tag{3.9}$$

Constraint (3.8) ensures that if production family f becomes active on unit j at time interval $t-1$ it stays active unless determined to finish at time interval t. Constraint (3.9) ensures a product family f can only finish if it was active at the previous time interval.

$$\sum_{f} Fa_{fjt} \leq 1 \qquad \forall j,\ t \tag{3.10}$$

$$\sum_{t} Ff_{fjt} \leq 1 \qquad \forall j,\ f \in F_j \tag{3.11}$$

Constraint (3.10) states that only one production family f may be active at time interval t on unit j, this restrict tasks $i \notin f$ from occurring when product family f is active.

Constraint (3.11) is key to the construction of the production policy. Restricting the number of times a processing family is allowed to finish to 1, restricts all active tasks $i \in I_f$ to occur as a single group within the production schedule. This group is allowed to span multiple time intervals.

It is noted that Fa_{fjt} can be specified as continuous on $[0, 1]$. If it is optimal for a product family to be active at time interval t, Fa_{fjt} will be assigned a value of 1 through constraint (3.7). Conversely, Ff_{fjt} will assume values of 1 to counter Fa_{fjt} through constraint (3.8). It is possible for Fa_{fjt} and Ff_{fjt} to assume values between 0 and 1, but this will not pose additional restrictions on the optimal solutions.

Production Grouping

As discussed in Section 1.2 product families with similar material compositions are grouped together for product cross contamination concerns. Production tasks i are grouped into 6 production groups which are indexed by g and I_g is given as the set of tasks i within production group g. Similar to the above constraints the activity of production group g will be tracked at each time interval to restrict the tasks which can be active in production group g. Ga_{gt} is introduced and defined as the activity level of production grouping g at time interval t. Equipment index j is purposely excluded from Ga_{gt} as our definition of production groups is defined across all units, although it may be defined over sub-groups of equipment units if desired. If any task $i \in I_f$ is active at time interval t the appropriate production group must also be active.

$$Ga_{gt} \geq \sum_{i \in I_g \cap I_j} \sum_{t'=t}^{t-pt_i+1} W_{ijt'} \qquad \forall g,\ j,\ t \tag{3.12}$$

Constraint (3.12) forces the activity variable Ga_{gt} to equate to 1 if a processing task i in production group g is beginning or actively processing at time t.

$$Ga_{gt} \geq Ga_{g,t-1} - W_{ijt} \qquad \forall g,\ j,\ i \in I_j \cap I_c,\ t \tag{3.13}$$

$$\sum_g Ga_{gt} \leq 1 \qquad \forall t \tag{3.14}$$

Constraint (3.13) represents the requirement that if a production group becomes active at time t it should stay active until a process wide cleaning task $i \in I_c$ occurs. Constraint (3.14) restricts only one production grouping to be active at any time t.

Ga_{gt} is specified as continuous on $[0,1]$. Through constraint (3.12) Ga_{gt} will assume values of 1 if any task $i \in I_g$ is active at event point n. It is possible for Ga_{gt} to assume values between 0 and 1, but this will not pose additional restrictions on the optimal solution. The above formulation can also be used to enforce the grouping of tasks that use particular utility resources, or for other heuristic reasons. It is noted that the above constraint can be used to enforce production groups on different units through the inclusion of index j, or sub-groups of units j, in variable Ga_{gt}.

Task Occurrence Restrictions

The processing policies of the target facility state that production tasks should not be duplicated within the production schedule. As the processing times of tasks i are calculated prior to optimization, for makespan minimization, this policy requirement can be expressed as constraints:

$$\sum_{t} \sum_{j \in K_i} W_{ijt} \leq \kappa_i \qquad \forall i \notin I_c \tag{3.15}$$

Constraint (3.15) limits the number of times processing task i is allowed to start within the scheduling horizon to κ_i times. The above constraints are not enforced for plant wide cleaning tasks $i \in I_c$, as these tasks are required to recur multiple times in a production schedule. K_i is the set of units j that can perform task i.

3.1.3 Objective Functions and Model Definition

Only the objective of makespan minimization will be considered in the discrete time model. Variable Tms is introduced to represent the makespan of the optimized production schedule. To relate T_{ms} to the completion time of the final processing task the following constraints are imposed:

$$T_{ms} \geq W_{ijt}\left(t + pt_i - 1\right) \qquad \forall i,\ j,\ t \tag{3.16}$$

The makespan minimization objective is then defined as:

$$\min\ T_{ms} \tag{3.17}$$

The discrete time model formulation is given as constraints (3.1) to (3.16) and objective function (3.17) and is referenced as model M_D

3.2 General Continuous Time Model

As in the discrete state-task network, states are represented through index s and account for all raw, intermediate and product materials. Processing tasks are indexed as i and processing units are indexed as j. Set I_j is introduced as the set of tasks i which can be performed on unit j. A key feature of the continuous-time formulation is the postulation of a set of event points, denoted through index $n \in N$, whose timing is not specified a priori but determined as a result of the optimization. N represents the set of event points and H defines the length of the scheduling horizon. The exact timing of each event point is tracked through the use of variable T_n.

Time Ordering Constraints

The ordering and relative timing of event points n is tracked through variables T_n and the given constraints.

$$T_{n=1} = 0 \tag{3.18}$$

$$T_{n=|N|} = H \tag{3.19}$$

$$T_{n-1} \leq T_n \qquad \forall n \tag{3.20}$$

The first event point must correspond to the start of the horizon, while the last event point must correspond to the horizon length, as given in constraint (3.18) and (3.19), respectively. H is a parameter defining the total horizon length. The timing of the current event point must be equal to or greater than the timing of the previous event point, as given in constraint (3.20).

Assignment Constraints

A key aspect of the proposed model is the decoupling of tasks i from equipment units j. If a task is capable of operation in multiple units it is modelled as one task per unit. Three binary variables are introduced for the assignment of task i to event point n.

$Ws_{in} = 1$ if task i begins at event point n
$Wp_{in} = 1$ if task i is being processed at event point n
$Wf_{in} = 1$ if task i finishes at or before event point n

Three additional auxiliary binary variables are introduced to aid the construction of the assignment constraints.

$Zs_{jn} = 1$ if a task in I_j begins in unit j at event point n
$Zp_{jn} = 1$ if a task in I_j is being processed in unit j at event point n
$Zf_{jn} = 1$ if a task in I_j processing to unit j, finishes at or before event point n

Disjunctive programming notation is used to discuss and derive the proper assignment constraints. $\vee$ represents logical operator *Or*, and notation such as $\underset{i \in I_j}{\vee}$ states or any element in set $i \in I_j$. $\Leftrightarrow$ represents the *If and Only If* operator, which states the statement is true if and only if all referenced logic is equivalently true. $\Rightarrow$ represents the *If Then* operator, when the original operator is true it implies truth in the target operator. $\neg$ is logical operator *Not*, stating the statement is true when the target logic is false. As an example the statement $\neg Ws_{in}$ is true when variable Ws_{in} is false.

Zs_{jn} is equivalently equal to 1 if and only if one of the tasks that can be assigned to unit j, is assigned to begin in unit j at event point n. The above condition can be represented as expression (A), a logical condition in which the binary variables are treated as Boolean variables. Similarly, Zf_{jn} is equal to 1 if and only if one task that can be assigned to unit j, is assigned to finish at or before event point n, which is given as logical condition (B).

$$Zs_{jn} \Leftrightarrow \bigvee_{i \in I_j} Ws_{in} \qquad \forall j,\ n \tag{A}$$

$$Zf_{jn} \Leftrightarrow \bigvee_{i \in I_j} Wf_{in} \qquad \forall j,\ n \tag{B}$$

Also, a task can be assigned to start in unit j at event point n, only if there is no other task being processed in unit j at event point n, and is expressed as logical condition (C).

$$Zs_{jn} \Rightarrow \neg Zp_{jn} \tag{C}$$

The binary variable Zp_{jn} can be represented as integer expression (D).

$$Zp_{jn} = \sum_{n' < n} Zs_{jn'} - \sum_{n' \leq n} Zf_{jn'} \qquad \forall j,\ n \tag{D}$$

The core assignment constraint is derived from logical conditions (A), (B) and (C).

$$\sum_{n' \leq n} \sum_{i \in I_j} (Ws_{in'} - Wf_{in'}) \leq 1 \qquad \forall j,\ n \tag{3.21}$$

Logical condition (C) can be transformed into the equivalent mixed integer assignment constraint (3.21). It ensures that a task cannot start unless all tasks that have previously commenced on that unit have finished. The complete derivation of constraint (3.21) is given in Appendix A. The following additional constraints are imposed:

$$\sum_{i \in I_j} Ws_{in} \leq 1 \qquad \forall j,\ n \tag{3.22}$$

$$\sum_{i \in I_j} Wf_{in} \leq 1 \qquad \forall j,\ n \tag{3.23}$$

$$\sum_{n} Ws_{in} = \sum_{n} Wf_{in} \qquad \forall i \tag{3.24}$$

Constraint (3.22) ensures that only one task can start in unit j at event point n, while constraint (3.23) enforces that only one task can finish in unit j at event point n. Constraint (3.24) states that all tasks that start must finish. In addition, no task is allowed to finish at the first event point ($Wf_{i,1} = 0$) and no task is allowed to begin at the end of the scheduling horizon ($Ws_{i,n=|N|} = 0$).

Start, Processing and Finish Time Constraints

Variable Ts_{in} is defined as the time that task i begins processing at event point n, while Tf_{in} is given as the finishing time of task i at event point n. The total processing time of task i at event point n is represented by variable Tp_{in}. These variables are related through the following constraints:

$$Tp_{in} = \alpha_i Ws_{in} + \beta_i Bs_{in} \qquad \forall i, n \qquad (3.25)$$

$$Tf_{in} \leq Ts_{in} + Tp_{in} + H\left(1 - Ws_{in}\right) \qquad \forall i,\ n \qquad (3.26)$$

$$Tf_{in} \geq Ts_{in} + Tp_{in} - H\left(1 - Ws_{in}\right) \qquad \forall i,\ n \qquad (3.27)$$

Constraint (3.25) defines the total processing time of task i as a combination of the fixed (α_i) and variable processing rates (β_i), where Bs_{in} represents the batch size of task i started at event point n. The finishing time of a task must be equal to the start time plus the processing time of task i, and is expressed through constraints (3.26) and (3.27).

Time Matching Constraints

The finish time of task i will remain unchanged until the next occurrence of task i, as enforced through constraint (3.28). Constraint (3.29) ensures that the jump in the processing time when task i recurs is greater than the processing time of task i. Although not explicitly needed for matching the finishing time, its inclusion leads to smaller branch-and-bound trees and shorter computational times [Maravelias and Grossmann, 2003b].

$$Tf_{in} - Tf_{i,n-1} \leq H\ Ws_{in} \qquad \forall i,\ n \qquad (3.28)$$

$$Tf_{in} - Tf_{i,n-1} \geq Tp_{in} \qquad \forall i,\ n \qquad (3.29)$$

The start time of any task i must correspond to the timing of an event point n, and must be greater than or equal to the finishing time of any task i currently finishing at event point n. When a task i produces a material with a zero-wait storage policy, the finishing time of this task must coincide with the beginning of the next event point n. These requirements are enforced through the constraints below, where ZW in constraint (3.32) represents the set of tasks i that have a zero-wait storage policy.

$$Ts_{in} = T_n \qquad \forall i,\ n \tag{3.30}$$

$$Tf_{in-1} \leq T_n + H\,(1 - Wf_{in}) \qquad \forall i,\ n \tag{3.31}$$

$$Tf_{in-1} \geq T_n - H\,(1 - Wf_{in}) \qquad \forall i \in ZW,\ n \tag{3.32}$$

Batch Size and Material Balance Constraints

Batch size variables are introduced to track the size of a batch throughout the processing steps within the scheduling horizon. Bs_{in}, Bp_{in} and Bf_{in} represent respectively the batch size of task i that starts at event point n, that is in process at event point n, and that finishes at or before event point n. The following constraints are imposed:

$$B_i^{min} Ws_{in} \leq Bs_{in} \leq B_i^{max} Ws_{in} \qquad \forall i,\ n \tag{3.33}$$

$$B_i^{min} Wf_{in} \leq Bf_{in} \leq B_i^{max} Wf_{in} \qquad \forall i,\ n \tag{3.34}$$

$$B_i^{min}(\sum_{n'<n} Ws_{in'} - \sum_{n'\leq n} Wf_{in'}) \leq Bp_{in} \qquad \forall i,\ n \tag{3.35}$$

$$B_i^{max}(\sum_{n'<n} Ws_{in'} - \sum_{n'\leq n} Wf_{in'}) \geq Bp_{in} \qquad \forall i,\ n \tag{3.36}$$

$$Bs_{i,n-1} + Bp_{i,n-1} = Bp_{in} + Bf_{in} \qquad \forall i,\ n \tag{3.37}$$

Every batch of task i that begins at event point n must lie within the minimum and maximum batch sizes, as described by constraint (3.33). Constraint (3.34) upholds the restriction that any batch of task i that finishes processing at event point n must also lie within the minimum and maximum batch sizes. Parameters B_i^{min} and B_i^{max} represent the minimum and maximum batch size of task i, respectively. The batch in progress must also lie within the maximum and minimum range and is given as constraints (3.35) and (3.36). Constraint (3.37) is used to ensure that the batch size of any task i remains constant from beginning to end.

The amount of material state s consumed by production task i at event point n is tracked through variable B^I_{isn}. Similarly, variable B^O_{isn} is introduced as the amount of state s produced by task i at event point n. The constraints below ensure that these variables are equal to the equivalent proportion of any batch that is beginning and finishing at event point n.

$$B^I_{isn} = \rho_{is} Bs_{in} \qquad \forall i,\ n,\ s \in SI_i \tag{3.38}$$

$$B^O_{isn} = \bar{\rho}_{is} Bf_{in} \qquad \forall i,\ n,\ s \in SO_i \tag{3.39}$$

SO_i and SI_i represent the set of states s that are produced and consumed by task i, respectively. Parameters $\bar{\rho}_{is}$ and ρ_{is} are the specified proportions of state s generated and consumed by production task i respectively.

Variables B^I_{isn} and B^O_{isn} are also bounded relative to the maximum batch size of task i, enforced through

$$B^I_{isn} \leq \rho_{is}\ B^{max}_i\ Ws_{in} \qquad \forall i,\ n,\ s \in SI_i \tag{3.40}$$

$$B^O_{isn} \leq \bar{\rho}_{is}\ B^{max}_i\ Wf_{in} \qquad \forall i,\ n,\ s \in SO_i \tag{3.41}$$

Material Balance Constraints

The material balance of the production system is given as

$$S_{sn} + SS_{sn} = S_{s,n-1} + \sum_{i \in O_s} B^O_{isn} - \sum_{i \in I_s} B^I_{isn} \qquad \forall s,\ n \tag{3.42}$$

The constraint states that the current quantity of state s (S_{sn}) plus product sales (SS_{sn}) is equal to the previous quantity of state s plus the amount of state s produced minus the amount consumed. I_s and O_s represent the sets of tasks i that consume and produce state s respectively. Sales of product state s are required to be greater than any specified demands for the scheduling horizon. The following constraints are enforced:

$$\sum_n SS_{sn} \geq D_s \qquad \forall s \tag{3.43}$$

Sequence Dependent Changeovers

In the model formulation of Maravelias and Grossmann [2003b], sequence dependent changeovers require the addition of extra event points to accurately model material release from processing tasks. Under certain circumstances, a modelling simplification can be used to remove the requirement for additional event points. For example, the target facility is modelled as single-stage and product states require no further processing. Therefore it is possible to postpone the occurrence of the next event point while maintaining material transfer policies. The modified sequence dependent changeover constraints are given as:

$$Ts_{i'n} \geq Tf_{i,n-1} + cl_{ii'} - H(1 - Ws_{i'n}) \qquad \forall j,\ n,\ i \in I_j,\ i' \in I_j | cl_{ii'} > 0,\ n > 1 \tag{3.44}$$

Constraint (3.44) ensures that the start time of task i' occurs at least $cl_{ii'}$ hours from the completion of task i, the constraints are enforced only if task i' occurs at event point n. Parameter $cl_{ii'}$ is defined as the changeover time from task i to i'. The above constraints effectively postpone the timing of the completion of task i, and no additional event points are required to model sequence dependent changeovers.

Tightening Constraints

Several tightening inequalities proposed in Maravelias and Grossmann [2003b] are used to improve the computational performance of the continuous-time model. To track the occurrence of process changeovers variable $CO_{ii'n}$ is introduced and defined as 1 if a changeover from task i to i' occurs at event point n. The following constraints are imposed:

$$CO_{ii'n} \geq Wf_{in} + Ws_{i'n} - 1 \qquad \forall j,\ i \in I_j,\ i' \in I_j,\ 1 < n < |N|,\ cl_{ii'} > 0 \tag{3.45}$$

$$\sum_{i \in I_j} \sum_{i' \in I_j} CO_{ii'n} \leq 1 \qquad \forall j,\ n \tag{3.46}$$

Constraint (3.45) equates $CO_{ii'n}$ to 1 if task i finishes and task i' begins at event point n. The constraint is a modification from that originally proposed; $Wf_{i,n}$ is changed from $Wf_{i,n-1}$. This modification is done to remain consistent with the treatment of sequence dependent changeovers in the current formulation. Constraint (3.46) ensures the sum of changeover indicator variables is less than or equal to 1 at event point n for every unit j. Although not part of the original formulation, the constraint leads to improved computational performance. Having defined the changeover indicator variable, the tightening inequalities are given as:

$$\sum_{i \in I_j} \sum_{n} Tp_{in} + \sum_{n} \sum_{i \in I_j} \sum_{i' \in I_j} cl_{ii'} CO_{ii'n} \leq H \qquad \forall j \qquad (3.47)$$

$$\sum_{i \in I_j} \sum_{n \leq n'} Tp_{in'} + \sum_{n' > n} \sum_{i \in I_j} \sum_{i' \in I_j} cl_{ii'} CO_{ii'n'} \leq H - T_n \qquad \forall j,\ n \qquad (3.48)$$

$$\sum_{i \in I_j} \sum_{n' \leq n} (\alpha_i \; Wf_{in'} + \beta_i \; Bf_{in'}) + \sum_{n' \leq n} \sum_{i \in I_j} \sum_{i' \in I_j} cl_{ii'} CO_{ii'n'} \leq T_n \qquad \forall j,\ n \qquad (3.49)$$

Constraint (3.47) states that the total processing time of tasks on unit j plus the sum of the required changeover time should not exceed the total horizon time. Constraint (3.48) implies that the total processing time plus the required changeover time on unit j after event point n must be less than the remaining horizon time. The total processing time of tasks finished at or before event point n plus the required changeover time on unit j must also be less than the timing of event point n, as enforced through constraint (3.49). It is noted that $CO_{ii'n}$ is treated as continuous on [0,1]. $CO_{ii'n}$ will assume values of 1 if a changeover from task i to i' occurs at event point n. It is important to note variable $CO_{ii'n}$ is not restricted to strictly assume values 0 and 1, but can assume values in between. However, $CO_{ii'n}$ is bounded by participation in constraints (3.46) to (3.49), and can only assume values in between [0,1] if it does not pose additional restrictions on the optimal solution.

3.2.1 Extension To Plant Wide Cleaning Tasks

A plant wide cleaning operation must occur within 36 hours of the start of process operations and thereafter within 36 hours from the previous plant wide cleaning operation. In this work one can consider the plant wide cleaning operations as a type of maintenance that is carried out periodically, with a recurring interval no greater than Γ hours. Each occurrence of a plant wide cleaning operation is represented with a unique task which belongs to the set of all tasks $i \in I$. The duration of the plant wide cleaning task is prescribed by variable Tp_{in} through parameter α_i as in constraint (3.25). Let I_c represent the set of plant wide cleaning tasks. The planned maximum start time of plant wide cleaning task $i \in I_c$ is specified through parameter TC_i and occurs at intervals according to plant policy. Gantt chart A in

Figure 3.2 illustrates these maintenance start time limits. It is noted that tasks $i \in I_c$ are process wide and as such each task appears in sets I_j.

As the timing of event points is unknown prior to optimization, the formulation must be free to assign plant wide cleaning tasks to any event point n in the scheduling horizon. This is accomplished with the introduction of binary variable Y_{in}, which is defined as 1 if plant wide cleaning task i begins at event point n. It may be optimal to schedule such maintenance before the maximum time limit occurs; this can be accommodated through use a backward slack variable $\hat{T}_{in}$. $\hat{T}_{in}$ represents the backward timing slack from the specified planned maximum start time of task i. If plant wide cleaning task i occurs at event point n, then the timing of the current event point must equate to the timing of the plant wide cleaning operation:

$$T_n = TC_i \, Y_{in} + \bar{T}_{in} - \sum_{n' \leq n} \sum_{i' \leq i} \hat{T}_{i'n'} \qquad \forall i \in I_c, \ n \tag{3.50}$$

$$0 \leq \bar{T}_{in} \leq H_c \, (1 - Y_{in}) \qquad \forall i \in I_c, \ n \tag{3.51}$$

$$0 \leq \hat{T}_{in} \leq \theta_c \, Y_{in} \qquad \forall I \in I_c, \ n \tag{3.52}$$

Constraint (3.50) represents the timing assignment constraint and includes the allowance of back slack on the timing of plant wide cleaning tasks through $\hat{T}_{in}$. We draw the reader's attention to several key points. TC_i enforces the maximum start time of plant wide cleaning task i if no backward timing slack is allocated; this is shown in Gantt chart A of Figure 3.2. If it is determined the optimal start time of the cleaning task is prior to TC_i, the double summation term enforces a maximum of Γ processing hours between subsequent cleaning tasks. The first summation is over all event points, and the second is over the set of ordered plant wide cleaning tasks. Only if event point n' corresponds to a plant wide cleaning task, does $\hat{T}_{i'n'}$ contribute to the sum; it is zero otherwise through constraint (3.52). Scalar H_c is the upper bound of the slack necessary to relax constraint (3.50) and is given as $H + \sum_{i \in I_c} \theta_i$. $\bar{T}_{in}$ relaxes constraint (3.50) if plant wide cleaning task c does not occur at event point n.

The plant wide cleaning task formulation is illustrated in Figure 3.2. Chart A is constructed from the specified maximum processing time between plant wide cleaning tasks. Assume that the first plant wide cleaning task occurs at event point 2 and $\hat{T}_{12}$ hours prior to TC_1, shown in Figure 3.2B. TC_2 is now no longer within Γ hours of the completion of the first plant wide cleaning task. To maintain the integrity of the maintenance timing requirement, the timing of the second plant wide cleaning task is shifted backward to begin no later than $TC_2 - \hat{T}_{12}$ hours, illustrated in Figure 3.2C. A further slack may be applied to the second plant wide cleaning task, and the construction continued.

The following additional constraints are imposed:

$$\sum_n Y_{in} = 1 \qquad \forall i \in I_c \tag{3.53}$$

$$Y_{in} \leq Ws_{in} \qquad \forall i \in I_c,\ n \tag{3.54}$$

$$Y_{in} \geq Ws_{in} \qquad \forall i \in I_c,\ n \tag{3.55}$$

Every process wide cleaning task $i \in I_c$ is required to occur at an event point in the scheduling horizon, as enforced through constraint (3.53). Constraints (3.54) and (3.55) ensure that process wide cleaning tasks $i \in I_c$ only occur in accordance with binary Y_{in}.

It is also necessary to ensure that the final plant wide cleaning task finishes within Γ hours of the final horizon time. The variable Tfc is introduced to represent the finishing time of the final plant wide cleaning task. The following constraints are enforced:

$$Tfc \leq Tf_{in} + H\,(1 - Y_{in}) \qquad \forall i \in |I_c|,\ n \tag{3.56}$$

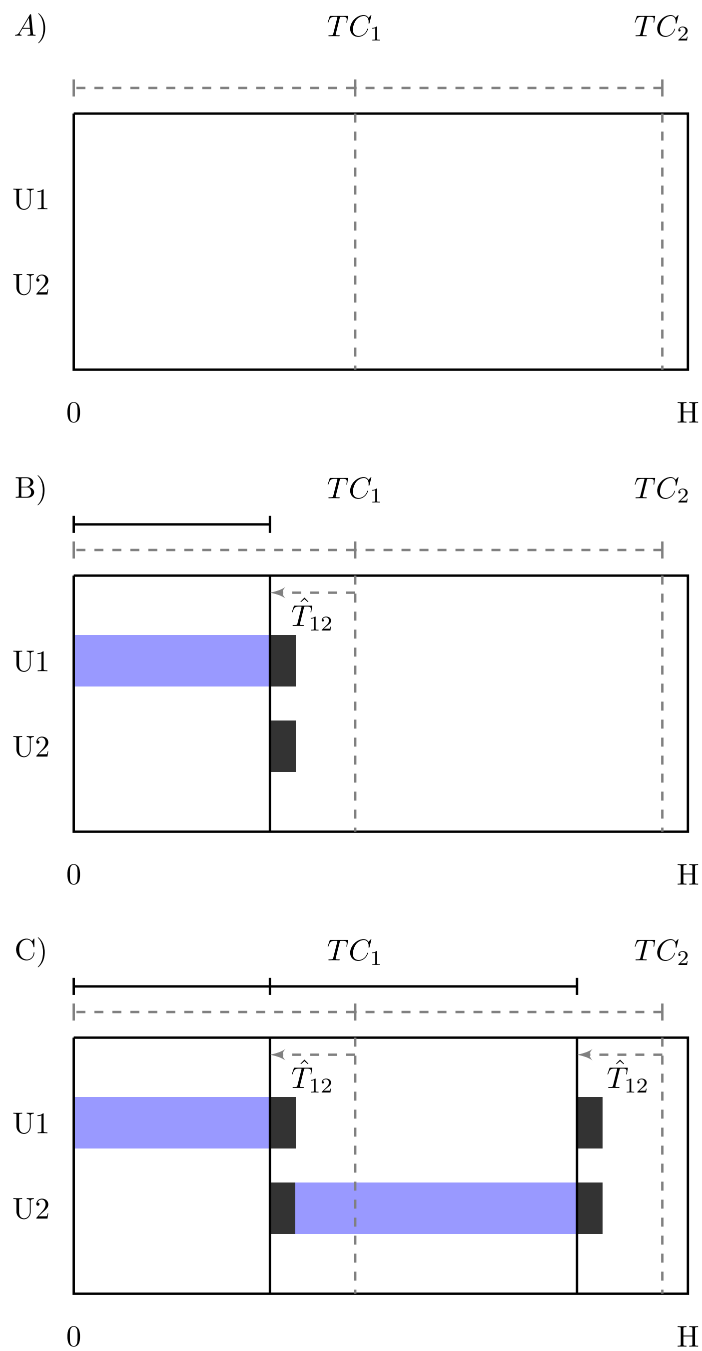
$$Tfc \geq Tf_{in} - H\,(1 - Y_{in}) \qquad \forall i \in |I_c|,\ n \tag{3.57}$$

$$H - Tfc \leq \Gamma \tag{3.58}$$

Constraints (3.56) and (3.57) enforce variable Tfc to equate to the finishing time of the final plant wide cleaning task. Constraint (3.58) enforces the final cleaning task to occur within Γ hours of the final horizon time.

It is noted that a similar methodology can be used to extend the global event continuous time framework with respect to general maintenance events with various timing requirements. Additional extensions to address maintenance events with fixed and flexible timing requirements is given in Appendix D. The extension can be used to concurrently optimize the timing of maintenance and processing tasks in multi-purpose batch plants.

Figure 3.2: Timeline of recurring cleaning events, A) Maximum start time limits of recurring maintenance events, B) Optimized timing of first maintenance event ($T_{12} > 0$), C) Optimized timing of second maintenance event.



3.2.2 Extension To Production Policies

The target facility follows three sets of production policies related to the relative ordering and occurrence of processing tasks within the production schedule. These are: i) all active tasks within a product family must be run as a group in the production schedule, ii) product families in a production group must be run together in the schedule, and must be separated by a plant wide cleaning task, and iii) no duplication of production tasks is allowed.

Product Family Grouping

As processing tasks in a product family all use the same material formulation it is required by plant policy to group such tasks together within the scheduling horizon. The product families are introduced as index f and I_f is given as the set of tasks i that belong to product family f. $f \in F_j$ is introduced as the set of product families f capable of production on unit j. All active tasks $i \in I_f$ are required to be run as a group within the production schedule, which is accomplished through restriction of the tasks which can be active on unit j at event point n. This is accomplished through variables Fa_{fjn} and Ff_{fjn}, which are introduced to track the activity of product families and are defined as follows: Fa_{fn} is equal to 1 if product family f is active or processing on unit j at event point n and Ff_{fjn} is equal to 1 if product family f is finishing on unit j at or before event point n. Product family f must be active at event point n for any tasks $i \in I_f$ to be processing. The following constraints are imposed:

$$Fa_{fjn} \geq Ws_{in} \qquad \forall j,\ f \in F_j,\ i \in I_f,\ n \tag{3.59}$$

$$Fa_{fjn} \geq \sum_{n'<n} Ws_{in'} - \sum_{n' \leq n} Wf_{in'} \qquad \forall j,\ f \in F_j,\ i \in I_f,\ n \tag{3.60}$$

Constraints (3.59) and (3.60) enforce product family f to become active if task $i \in I_f$ is starting or processing on unit j at event point n, respectively.

$$Fa_{fjn} \geq Fa_{fj,n-1} - Ff_{fjn} \qquad \forall j,\ f \in F_j,\ n \tag{3.61}$$

$$Ff_{fjn} \leq Fa_{jf,n-1} \qquad \forall j,\ f \in F_j,\ n \tag{3.62}$$

$$\sum_{f \in F_j} Fa_{fjn} \leq 1 \qquad \forall j,\ n \tag{3.63}$$

Constraint (3.61) ensures that if product family f becomes active on unit j at event point $n-1$, it remains active until determined to finish. A product family can only finish if it was previously active, as enforced through 3.62. Constraint (3.63) states that only one production family f may be active on unit j at event point n.

$$\sum_{n} Ff_{fjn} \leq 1 \qquad \forall j,\ f \in F_j \tag{3.64}$$

Constraint (3.64) is key to the construction of the above production policy. By limiting Ff_{fjn} to be less than or equal to one we are restricting the number of times a product family may become active to one. Through the above formulation it is possible to ensure all active tasks $i \in I_f$ to occur as a single group which can span multiple event points. This also requires that no product family be active at the final event point ($Fa_{f,n=|N|} = 0$).

It is noted that Fa_{fjn} and $Ff_{fj,n}$ may be treated as continuous on [0,1]. Fa_{fjn} must assume values of 1 in integer solutions when a respective task i is active, and conversely Ff_{fjn} must assume a value of 1 to counter Fa_{fjn}. It is allowable for Fa_{fjt} and Ff_{fjt} to assume values in between 0 and 1, but this can only occur if it does not pose additional restrictions on the optimal solution. The aforementioned constraints can also be used to enforce the recurrences of a single type of batch task to occur as a group, or through enforcement of tasks $i \in I_f$ to occur as a group on one unit to indirectly restrict which tasks can occur on supporting units.

Production Grouping

A production group is defined as a group of tasks i that produce product materials with similar properties and use similar raw materials. As such, production groups share similar product quality and processing concerns. In this formulation production groups are represented through index g, and I_g is given as the set tasks i producing products within production group g. Such a grouping requirement can be accomplished in a similar manner to the product family grouping; variable Ga_{gn} is introduced to track the activity of production group g at event points n. It is noted that in this framework a production group is independent of processing equipment units due to the plant layout, although it is possible to have production groups on single units or groups of equipment units. Any task $i \in I_g$ can not be processing unless the respective production group is active. The following constraints are imposed:

$$Ga_{gn} \geq Ws_{in} \qquad \forall g,\ i \in I_g,\ n \tag{3.65}$$

$$Ga_{gn} \geq \sum_{n'<n} Ws_{in'} - \sum_{n'\leq n} Wf_{in'} \qquad \forall g,\ i \in I_g,\ n \tag{3.66}$$

Constraints (3.65) and (3.66) equate $Ga_{g,n}$ to 1 if a task i in production group g is currently starting or processing at event point n, respectively. It is noted that the above constraints are only enforced for tasks i which belong to a production group g. If any task i is not included in set $I_g \forall g$ then that task is not restricted to occur in any specific production group. In that regard these tasks $i \notin I_g \forall g$ can occur at any event point within the scheduling horizon. This might be advantageous for product formulations that contain generic raw materials and pose no restricts on other product families.

$$\sum_{g} Ga_{g,n} \leq 1 \qquad \forall n \tag{3.67}$$

Only one production group g may be active at event point n, as given through constraint (3.67).

$$Ga_{gn} \geq Ga_{g,n-1} - E_n \qquad \forall g,\ n \tag{3.68}$$

If production group g becomes active at event point $n-1$ it must remain active until a plant wide cleaning task occurs, as enforced by 3.68. E_n is used to account for the activity of plant wide cleaning tasks and is defined as 1 if a plant wide cleaning task begins at event point n.

$$E_n \geq Y_{in} \qquad \forall i \in I_c,\ n \tag{3.69}$$

$$E_n \leq \sum_{i \in I_c} Y_{in} \qquad \forall n \tag{3.70}$$

Constraints (3.69) and (3.70) force variable E_n to equate to 1 when any plant wide cleaning task i occurs at event point n and zero otherwise. As E_n is bounded both above and below by binary variables it may only assume values of 0 or 1 in a integer solution and as such may be treated as continuous on [0,1]. Ga_{gn} is also treated as continuous on [0,1] for reasoning similar to that proposed for variables Fa_{fjn} and Ff_{fjn}.

The production grouping constraints can also be used in multipurpose batch plants to: i) group tasks based on the types of resources utilized, or to indirectly restrict what utilities are used at a given moment, ii) separate the production of intermediate and product states, a process similar to campaign type modes of operation, and to iii) group tasks which may require additional supervision or testing to occur as a group such that personnel are utilized effectively.

Task Occurrence Restrictions

The processing policies of the target facility state that production tasks should not be duplicated within the production schedule. This requirement can be formulated as the addition constraints given below:

$$\sum_n Ws_{in} \leq \kappa_i \qquad \forall i \tag{3.71}$$

$$\sum_n Wf_{in} \leq \kappa_i \qquad \forall i \tag{3.72}$$

Constraints (3.71) and (3.72) limit the number of times processing task i is allowed to start and finish within the scheduling horizon respectively. κ_i is a parameter defining the number of times task i is allowed to start or finish.

Auxiliary Constraints

As the facility is modeled as single-unit and single-stage, the binary indicator variables can be related as follows:

$$Ws_{in} \leq Wf_{i,n+1} \qquad \forall j \in J^*,\ i \in I_j,\ n \tag{3.73}$$

Constraint (3.73) enforces that a task i must finish at the subsequent event point from which it started. This remains valid for a single-unit facility as no other processing tasks can be run concurrently and therefor no additional events points are needed to model the optimal schedule. Here J^* represents the set of units for which such a auxiliary constraint applies.

3.2.3 Objective Functions

Two objective functions are considered in the industrial case studies - makespan minimization and throughput maximization. Throughput maximization takes the following form:

$$\max \quad Z = \sum_{s} \sum_{n} SS_{sn} \tag{3.74}$$

Variable Z represents the total weight equivalent of all states s sold to market in the scheduling horizon. Constraint (3.74) defines the throughput maximization objective function as the maximization of all material sold to market. In addition, it is also common for a profit maximization objective to be considered in scheduling optimization. [Maravelias and Grossmann, 2003b]

Makespan minimization takes the following form:

$$\min \quad Tms \tag{3.75}$$

Tms represents the makespan of the scheduling period, given as the completion time of the final processing task. Tms must replace parameter H in constraints (3.19), (3.47), (3.48) and (3.58) in order to enforce it as the makespan of the schedule.

Model M_C is defined as constraints (3.18) to (3.73), using objective (3.74) or (3.75). The minimum number of event points needed for the model can be calculated prior to optimization as follows: First, as the plant is modelled as a single-stage single-unit process the number positive demand parameters will correspond to the number of tasks that must be conducted. Second, the minimum number of process wide cleaning tasks needed can be calculated from the horizon length, time limit between tasks and the duration of the cleaning tasks. The combination of the minimum number of processing tasks and plant wide cleaning tasks combine, plus one, to determine the minimum number of event points.

In addition, model sizes are reduced by restricting the binary variables that participate in the optimization in the following way: Let D^+ represent the set of all product states s with a positive demand parameter; then the following binary restriction is imposed for problems:

$$Ws_{in} = Wf_{in} = 0 \qquad \forall s \notin D^+,\ i \in O_s$$

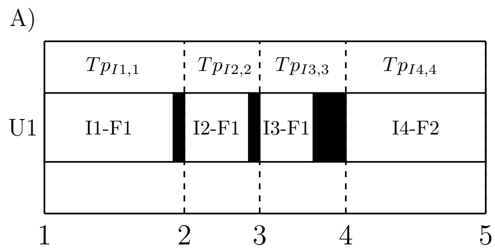
Such restrictions aid in reducing model sizes and improving the computational performance of the model.

3.3 Aggregate Continuous Time Model

The solution of large and complex continuous time scheduling models can be computationally very challenging. It was stated by Méndez *et al.* [2006] that the complex structure of continuous time models makes them useful for problems with a modest number of event points, referencing 15 event points as the possible upper bound. Lin *et al.* [2002] and Janak *et al.* [2006] present work on medium-term scheduling of batch-plants. The horizon is decomposed into smaller more manageable sub-horizons which are solved in an iterative solution procedure. Examination of the reduced horizon model complexity implicitly bounds the number of unit-event points between 10 to 20. The above observations are also present in the computational results presented by Shaik *et al.* [2006] where the authors compare the computational efficiency of continuous time modeling methodologies. Historically, the processing facility of this research produces in excess of 15 different products on a weekly basis. As the plant is modeled as single stage this implies that in excess of 15 event points will be required to model a production schedule, and may thus lead to computational difficulties.

Another important issue with MILP optimization is the existence of multiple equivalent solutions for a given problem. These equivalent solutions represent symmetry within the optimization problem caused by binary variables associated with indistinguishable objects. This causes mirroring in the solution procedure, as the algorithm is forced to explore sym-

Figure 3.3: Task–Family Aggregation, A) General continuous time mode, sequencing optimized between all tasks i, B) Aggregation of tasks $i \in I_f$ such that sequencing is optimized between product families



metrical reflections of multiple solutions during the branch-and-bound search leading to performance degradation. [Sherali and Smith, 2001] To circumvent this problem, model augmentation or reformulation is often necessary to remove or reduce the inherent symmetries. Within scheduling problems, the assignment binaries can be associated with indistinguishable objects; in various instances task sequence may be permuted without effect on the objective function. As problem size grows, the symmetry issues are compounded as multiple product pathways exist and equipment can be utilized in multiple ways at different times to produce the same output.

In the current study, the sequence of active production tasks i in product families f pose a symmetry issue. First, the processing tasks $i \in I_f$ are required to be sequenced as a group in the production schedule. Second, changeover durations between tasks $i \in I_f$ are identical for any product family. Therefore, the sequence of tasks i in a product family f is irrelevant and will not impact the objective function of the scheduling problem. Each possible sequence of active tasks $i \in I_f$ will be a symmetrical solution for the scheduling problem. It would be advantageous to remove these symmetrical solutions from the MILP problem to improve the computational performance of the scheduling optimization.

3.3.1 Model Reformulation - Task–Family Aggregation

One intuitive method to remove these inherent symmetrical solutions is to remove the need to sequence individual tasks $i \in I_f$. Tasks i can be aggregated into the respective product families f and sequencing can be conducted on the product families. To accomplish this two additional binary variables are introduced: Vs_{fn} is defined as 1 if product family f is begins processing at event point n, and zero otherwise. Vf_{fn} is defined as 1 if product family f finishes at or before event point n. Figure 3.3 presents a comparison between the general (A) and aggregate (B) continuous time modelling techniques. The general continuous time model requires tasks $i \in I_f$ to occur as a group, but each occurrence of a task requires an event point. The processing time of task i is given through variable Tp_{in}. The aggregate continuous time model sequences the product families at event points n and all active tasks $i \in I_f$ occur at this event point. The total processing time of the aggregate processing family is given through variable Tp_{fn}, which is a combination of the total processing time and changeover time of tasks $i \in I_f$.

It is noted that such an aggregation presents two assumptions: First, the sequence of the tasks $i \in I_f$ is no longer determined as part of the optimization and is left as a decision to the operators. Second, the aggregation of different tasks that occur at the same event point can interfere with material transfer policies and product due dates. However in the present study these assumptions do not pose a problem. Castro *et al.* [2008] present a aggregation technique that is similar in spirit to that used in the present study. Multiple occurrences of a batch processing task are aggregated into an explicit task with the number of batches accounted for through integer variables. The processing time of the explicit aggregation is determined as the combination of total batch processing and changeover times.

Event Point Ordering

The timing and ordering of event points remains unchanged and is repeated below.

$$T_{n=1} = 0 \tag{3.76}$$

$$T_{n=|N|} = H \tag{3.77}$$

$$T_{n-1} \leq T_n \qquad \forall n \tag{3.78}$$

Family Sequence Assignment

The sequencing is now conducted on product families as opposed to the individual tasks

$i \in I_f$. The assignment constraints are restated as:

$$\sum_{f \in F_j} \sum_{n' \leq n} \left(Vs_{fn'} - Vf_{fn'} \right) \leq 1 \qquad \forall j,\ n \qquad (3.79)$$

The main assignment constraint is now given as constraint (3.79), and is identical to the original assignment constraint except with a change of variables.

$$\sum_{f \in F_j} Vs_{f,n} \leq 1 \qquad \forall j,\ n \qquad (3.80)$$

$$\sum_{f \in F_j} Vf_{f,n} \leq 1 \qquad \forall j,\ n \qquad (3.81)$$

$$\sum_{f \in F_j} Vs_{f,n} = \sum_{f \in F_j} Vf_{f,n} \qquad \forall f \qquad (3.82)$$

Constraint (3.80) enforces that only one product family f can start on unit j at event point n. Similarly, constraint (3.81) enforces that only one product family f can end on unit j at or before event point n. All product families that begin must finish, as given through constraint (3.82).

The following additional constraints are imposed:

$$Ws_{i,n} \leq Vs_{f,n} \qquad \forall f,\ i \in I_f,\ n \qquad (3.83)$$

$$Wf_{i,n} \leq Vf_{f,n} \qquad \forall f,\ i \in I_f,\ n \qquad (3.84)$$

$$\sum_{n} Ws_{in} = \sum_{n} Wf_{in} \qquad \forall i \qquad (3.85)$$

Processing tasks $i \in I_f$ are allowed to occur only if family f is scheduled to begin at event point n, as enforced through constraint (3.83). Constraint (3.84) enforces that task i finishes at event point n if product family f is scheduled to end also. Constraint (3.85) ensures that all tasks i that start must also finish. As before, no product family is allowed to finish at the beginning of the horizon, $Vf_{f,1} = 0$, and no family is allowed to begin at the end of the horizon, $Vs_{f,n=|N|} = 0$. Similar restrictions are enforced on processing tasks, i.e. $Ws_{i,1} = 0$ and $Wf_{i,n=|N|} = 0$

Family Processing Time Constraints

The processing time of a product family is track through variable Tp_{fn} and represents the total processing time of product family f at event point n. The processing time of family group f is calculated as the sum of the processing times of tasks $i \in I_f$ plus the required changeover time between tasks. The changeover time within a production family is expressed as $\gamma_f(\sum_{i \in I_f} Ws_{in} - Vs_{fn})$, where γ_f is defined as the changeover time between tasks $i \in I_f$.

$$Tp_{fn} = \sum_{i \in I_f} (\gamma_f Ws_{in} + \beta_i Bs_{in}) - \gamma_f Vs_{fn} \qquad \forall f,\ n \qquad (3.86)$$

Constraint (3.86) defines the processing time of product family f at event point n to be equal to the variable processing time of tasks $i \in I_f$, plus the changeover time required within the product family. Through this definition it is possible to have multiple tasks $i \in I_f$ occur at the same event point, as the overall processing time is enforced explicitly. It is noted that fixed set-up times for tasks i (α_i) is not included in the above formulation. This is excluded for this problem, but can more generally be included in the formulation.

Start and Finishing Time Constraints

Variables Ts_{fn} and Tf_{fn} now represent the start, finishing time of product family f at event point n, and participate in the following constraints:

$$Tf_{fn} \leq Ts_{fn} + Tp_{fn} + H\,(1 - Vs_{fn}) \qquad \forall f,\ n \qquad (3.87)$$

$$Tf_{fn} \geq Ts_{fn} + Tp_{fn} - H\,(1 - Vs_{fn}) \qquad \forall f,\ n \qquad (3.88)$$

Constraints (3.87) and (3.88) enforce the finishing time of product family f to be equal to the start time of family f plus the total processing time of product family f.

$$Tf_{fn} - Tf_{f,n-1} \leq H\ Vs_{fn} \qquad \forall f,\ n \qquad (3.89)$$

$$Tf_{fn} - Tf_{f,n-1} \geq Tp_{fn} \qquad \forall f,\ n \qquad (3.90)$$

The finish time of product family f will remain unchanged until the next occurrence of family f, this is enforced through constraint (3.89). Constraint (3.90) ensures the jump in the finishing time when product family f recurs is greater than the processing time of family f.

$$Ts_{fn} = T_n \qquad \forall f,\ n \tag{3.91}$$

$$Tf_{f,n-1} \leq T_n + H\,(1 - Vf_{fn}) \qquad \forall f,\ n \tag{3.92}$$

As event points are common for all production resources, the start time of family f must be equal to the timing of the current event point. This enforces the global event point representation and is given as constraint (3.91). The start times of product family f must be greater than or equal to the previous finish time of that family, as enforced through constraint (3.92). It is noted that zero-wait material transfer policies can not be expressly defined due to the aggregation of processing tasks $i \in I_f$ to occur at a single event point.

Batch Sizing

All batch sizing and material state production and consumption constraints remain unchanged and are restated below.

$$B_i^{min} Ws_{in} \leq Bs_{in} \leq B_i^{max} Ws_{in} \qquad \forall i,\ n \tag{3.93}$$

$$B_i^{min} Wf_{in} \leq Bf_{in} \leq B_i^{max} Wf_{in} \qquad \forall i,\ n \tag{3.94}$$

$$B_i^{min}\Big(\sum_{n'<n} Ws_{in'} - \sum_{n'\leq n} Wf_{in'}\Big) \leq Bp_{in} \qquad \forall i,\ n \tag{3.95}$$

$$B_i^{max}\Big(\sum_{n'<n} Ws_{in'} - \sum_{n'\leq n} Wf_{in'}\Big) \geq Bp_{in} \qquad \forall i,\ n \tag{3.96}$$

$$Bs_{i,n-1} + Bp_{i,n-1} = Bp_{in} + Bf_{in} \qquad \forall i,\ n \tag{3.97}$$

$$B_{isn}^{I} = \rho_{is} Bs_{in} \qquad \forall i,\ n,\ s \in SI_i \tag{3.98}$$

$$B_{isn}^{O} = \bar{\rho}_{is} Bf_{in} \qquad \forall i,\ n,\ s \in SO_i \tag{3.99}$$

$$B_{isn}^{I} \leq \rho_{is}\ B_i^{max}\ Ws_{in} \qquad \forall i,\ n,\ s \in SI_i \tag{3.100}$$

$$B_{isn}^{O} \leq \bar{\rho}_{is}\ B_i^{max}\ Wf_{in} \qquad \forall i,\ n,\ s \in SO_i \tag{3.101}$$

Material Balance

The material balance constraints, as before, are

$$S_{sn} + SS_{sn} = S_{s,n-1} + \sum_{i \in O_s} B^O_{isn} - \sum_{i \in I_s} B^l_{isn} \qquad \forall s,\ n \qquad (3.102)$$

$$\sum_{n} SS_{sn} \geq D_s \qquad \forall s \qquad (3.103)$$

Sequence Dependent Changeovers

Sequence dependent changeover constraints are now based on the sequencing of production families. This will aid in model size reduction as the target facility has roughly 30 product families as opposed to roughly 50 product types.

$$Ts_{f'n} \geq Tf_{f,n-1} + sl_{ff'} - H(1 - Vs_{f'n}) \quad \forall n,\ j,\ f \in F_j,\ f' \in F_j | sl_{ff'} > 0,\ n > 1 \quad (3.104)$$

Constraint (3.104) states that family f' must start at least $sl_{ff'}$ hours after family f on unit j. Parameter $sl_{ff'}$ is given as the time necessary to clean a given unit if family f precedes family f'.

Tightening Constraints

The tightening constraints can now be restated to account for the model aggregation. $CO_{ff'n}$ is defined as 1 if a changeover from family f to f' at event point n has occurred, and is zero otherwise.

$$CO_{ff'n} \geq Vf_{fn} + Vs_{f'n} - 1 \qquad \forall j,\ f \in F_j,\ f' \in F_j,\ n \qquad (3.105)$$

$$\sum_{f \in F_j} \sum_{f' \in F_j} CO_{ff'n} \leq 1 \qquad \forall j,\ n \qquad (3.106)$$

Constraint (3.105) equates indicator variable $CO_{ff'n}$ to 1 if family f finishes and family f' begins at event point n. Constraint (3.106) is introduced to ensure only one changeover indicator variable is forced to 1 at event point n on unit j.

The tightening inequalities are now given as:

$$\sum_{f \in F_j} \sum_{n} Tp_{f,n} + \sum_{n} \sum_{f \in F_j} \sum_{f' \in F_j} sl_{ff'} CO_{ff'n} \leq H \qquad \forall j \tag{3.107}$$

$$\sum_{f \in F_j} \sum_{n \leq n'} Tp_{f,n'} + \sum_{n' > n} \sum_{f \in F_j} \sum_{f' \in F_j} sl_{ff'} CO_{ff'n'} \leq H - T_n \qquad \forall j,\ n \tag{3.108}$$

$$\sum_{f \in F_j} \sum_{n' \leq n} \left(\sum_{i \in I_f} (\gamma_f W f_{in'} + \beta_i B f_{in'}) - \gamma_f V f_{fn'} \right) + \sum_{n' \leq n} \sum_{f \in F_j} \sum_{f' \in F_j} sl_{ff'} CO_{ff'n'} \leq T_n \qquad \forall j,\ n \tag{3.109}$$

Constraint (3.107) states that the total processing time of all product families plus the required changeover time on unit j should not exceed the total horizon time. Constraint (3.108) implies that all future processing time plus the required changeover time on unit j after the current event point must be less than or equal the time remaining in the scheduling horizon. Constraint (3.109) states that the total processing time of product families finishing at or before event point n plus the required changeover time on unit j must be less than the timing of event point n.

3.3.2 Extension To Recurring Maintenance Events

The recurring process wide cleaning constraints remain unmodified and are repeated below. It is noted that maintenance tasks i which represent the recurring process wide cleaning operations will now need to be defined with a product family. This will allow for maintenance tasks i to be sequenced within the scheduling horizon in relation to the sequencing of the product families.

$$T_n = TC_i\, Y_{in} + \bar{T}_{in} - \sum_{n' \leq n} \sum_{i' \leq i} \hat{T}_{i'n'} \qquad \forall i \in I_c,\ n \tag{3.110}$$

$$0 \leq \bar{T}_{in} \leq H_c\,(1 - Y_{in}) \qquad \forall i \in I_c,\ n \tag{3.111}$$

$$0 \leq \hat{T}_{in} \leq \theta_c\, Y_{in} \qquad \forall i \in I_c,\ n \tag{3.112}$$

$$\sum_n Y_{in} = 1 \qquad \forall i \in I_c \tag{3.113}$$

$$Y_{in} \leq Ws_{in} \qquad \forall i \in I_c,\ n \tag{3.114}$$

$$Y_{in} \geq Ws_{in} \qquad \forall i \in I_c,\ n \tag{3.115}$$

$$Tfc \leq Tf_{in} + H\,(1 - Y_{in}) \qquad \forall i \in |I_c|,\ n \tag{3.116}$$

$$Tfc \geq Tf_{in} - H\,(1 - Y_{in}) \qquad \forall i \in |I_c|,\ n \tag{3.117}$$

$$H - Tfc \leq \Gamma \tag{3.118}$$

3.3.3 Extension To Production Policies

Product Family Grouping

The product family grouping constraints presented in the general continuous time scheduling are not needed in the aggregate reformulation model. This is due to the implicit occurrence of tasks $i \in I_f$ to occur when family f occurs, noting that multiple tasks $i \in I_f$ can occur at a single event point.

Production Grouping

The production grouping constraints previous stated remain unmodified and are presented below. It is noted that production groups g are defined as sets of tasks which can occur as a group within the scheduling horizon. These groups are separated by plant wide cleaning tasks in the production schedule. I_g is defined as the set of tasks i which must occur in production group g.

$$Ga_{g,n} \geq Ws_{in} \qquad \forall g,\ i \in I_g,\ n \tag{3.119}$$

$$Ga_{g,n} \geq \sum_{n'<n} Ws_{in'} - \sum_{n' \leq n} Wf_{in'} \qquad \forall g,\ i \in I_g,\ n \tag{3.120}$$

$$Ga_{gn} \geq Ga_{g,n-1} - E_n \qquad \forall g,\ n \tag{3.121}$$

$$\sum_{g} Ga_{g,n} \leq 1 \qquad \forall n \tag{3.122}$$

$$E_n \geq Y_{in} \qquad \forall i \in I_c,\ n \tag{3.123}$$

$$E_n \leq \sum_{i \in I_c} Y_{in} \qquad \forall n \tag{3.124}$$

Task Occurrence Restrictions

As previously stated, production tasks i may only occur once in a production schedule. The following constraints are enforced:

$$\sum_{n} Ws_{in} \leq \kappa_i \qquad \forall i \tag{3.125}$$

$$\sum_{n} Wf_{in} \leq \kappa_i \qquad \forall i \tag{3.126}$$

Constraints (3.125) and (3.126) restrict the number of times task i can begin processing at event point n, and end at or before event point n to κ_i.

Auxiliary Constraints

As the facility is modelled as single-unit and single-stage, the binary indicator variables can be related as follows:

$$Ws_{in} \leq Wf_{i,n+1} \qquad \forall j \in J^*,\ i \in I_j,\ n \tag{3.127}$$

Constraint (3.127) enforces that a task i must finish at the subsequent event point from which it started. This remains valid for a single-unit facility as no other processing tasks can be run concurrently and therefor no additional events points are needed to model the optimal schedule. Here J^* represents the set of units for which such an auxiliary constraint applies.

3.3.4 Objective Functions and Model Definition

Throughput maximization is achieved via objective function:

$$\max Z = \sum_{s} \sum_{n} SS_{sn} \tag{3.128}$$

Makespan minimization is unchanged and repeated as constraint (3.129).

$$\min \; Tms \tag{3.129}$$

When makespan minimization is the chosen objective function constraints (3.77), (3.107) and (3.108) must be modified by replacing H with Tms. The symmetry aggregation model is defined by constraints (3.76) to (3.127) and one of the objectives (3.128) or (3.129). The model is referenced as M_S.

Chapter 4

Scheduling Case Studies

4.1 Model Comparison Case Study

It is necessary to ensure that the results generated from models M_D, M_C and M_S are equivalent, so that the model comparisons can be judged accurately. A solution quality comparative case study is given in this section to compare the schedules produced from the three models. The data for the comparative example is presented in Appendix C; the horizon length is set to 40 hours. The optimized schedules of the three models are represented as the Gantt charts in Figure 4.1. Although the plant is modelled as a single-stage single-unit facility, it is reiterated that this is due to physical connection restrictions within the facility and that the plant consists of 4 main packaging units. Therefore Gantt charts are drawn with these 4 packaging units, and tasks occur on the given units respectively. It is noted that processing tasks are represented as horizontal bars and changeovers are shown as black bars at the end of processing tasks. Each processing task i belongs to a product family f, and the association to family f is represented by the color and pattern displayed within the horizontal bars of task i. Plant wide cleaning tasks are represented as black bars that occur over all equipment units, and are three hours in duration.

The solution to model M_C is displayed as the top Gantt chart in Figure 4.1. The sequencing, with respect to product families f, is observed as F9-F16-F23, followed by a plant wide cleaning (PWC) task and concluded with family F8. The plant wide cleaning task is started at 28.5 hours into the production schedule. It is observed a total of 2.25 hours of changeover time is accumulated in the production schedule and is comprised of five 0.25 hour changeovers, and a 1 hour changeover. The 1 hour changeover is needed for transition between product family F9-F16 and occurs between tasks on units U1-U3. Four of the 0.25 hour changeovers are between tasks within the same product family; in family F9 this is the transition between tasks on units U3-U2 and U2-U1 while in family F16 this is the transition between tasks on units U3-U1 and U1-U2. The fifth 0.25 changeover is in transition from family F16-F23 (U2-U4). Similarly, solutions to models M_S and M_D are given as the middle and bottom Gantt chart respectively. The total amount of changeover time remains the same in all solutions, but the sequence of tasks and product families is different. All Gantt charts represent equivalent optimal solutions to the problem. It is noted that no production groups are active in the above solutions, this is due to the fact that all the above processing tasks do not belong to the set $i \in I_g$.

The computational results of models M_C, M_S and M_D for the case study are given in Table 4.1. The optimized makespan of 36.75 hours was proven optimal by both continuous time models. However, the discrete time model was left with an optimality gap of 0.68% after 3,600 CPUs. Model M_S exhibited the best computational performance, proving optimality in 1.662 CPUs after exploring 75 nodes. Model M_C proved optimality in 34.98 CPUs after exploring 4,188 nodes, while model M_D found the optimal solution but failed to prove optimality. The poor computational performance of the discrete time model may be attributed to the large model size, a consequence of the short time intervals required to represent the shortest changeover.

4.2 Scheduling Case Study 1

The first industrial case study is derived from historical production week “X” and requires 21 production targets to be met within a scheduling horizon of 108 hours ($H = 108$). The 21 processing tasks can be grouped into 11 product families. Product families F2, F26 and F15 are required to be processed in product group 3, while product families F14 and F28 belong to production group 2. Two plant wide cleaning tasks (PWC) are specified to occur within the scheduling horizon. The maximum planned start time for the first cleaning task

Figure 4.1: Solution quality case study Gantt charts: general continuous model (top), aggregate continuous model (middle), and discrete model (bottom)

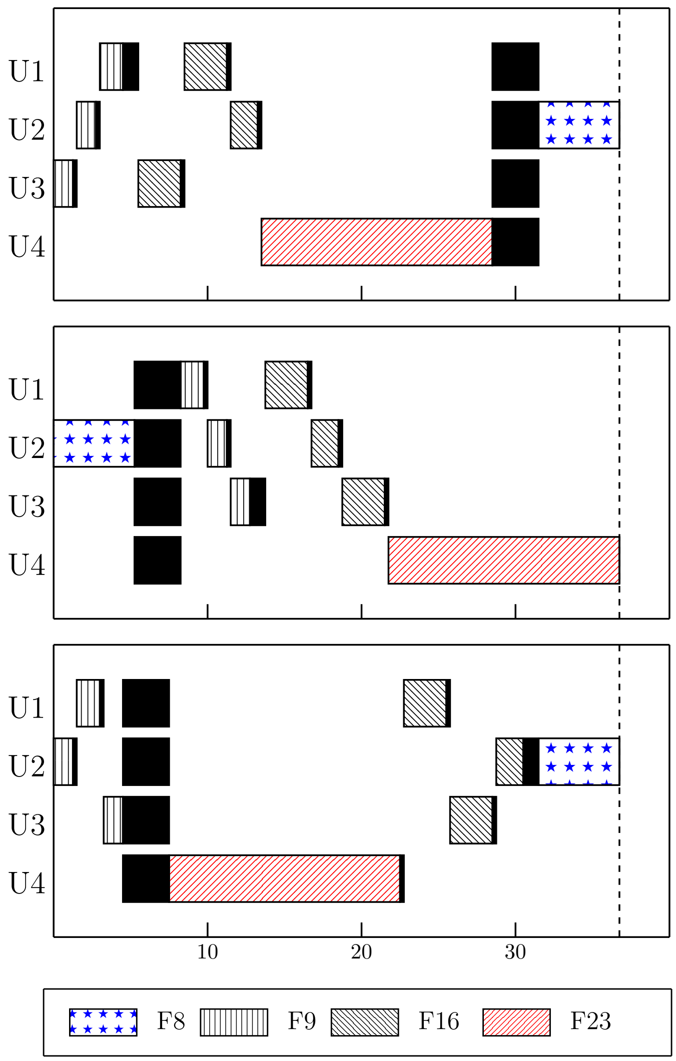


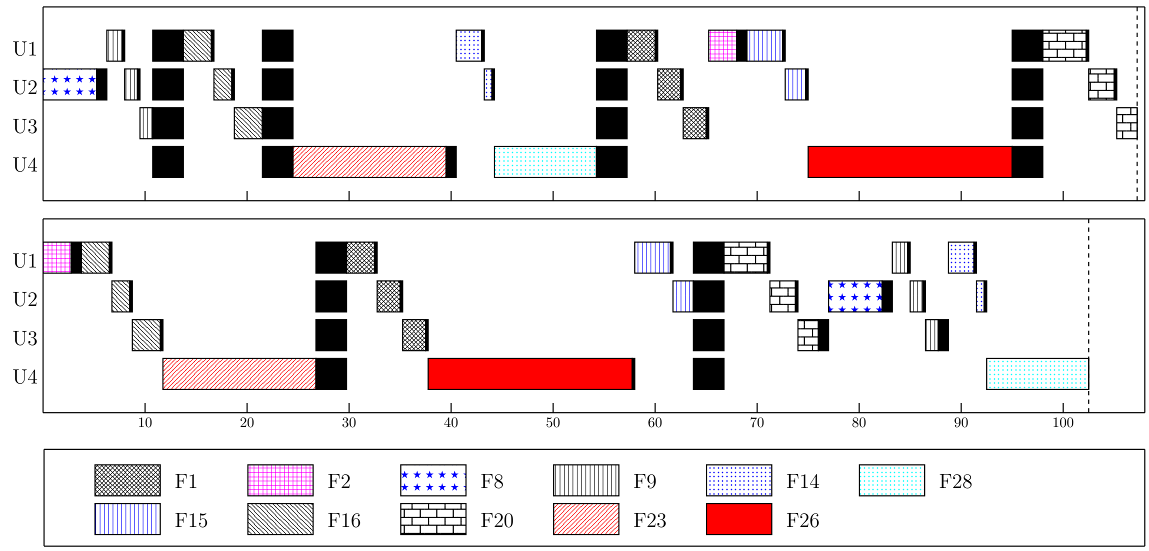
Table 4.1: Solution quality comparisons of models M_D, M_C and M_S.

	M_D (Δt=0.25hr)	M_C	M_S
Event Points (Time Periods)	160	10	6
Constraints	51,780	31,786	17,238
Binary Variables	1,449	190	186
Continuous Variables	43,149	46,533	14,673
LP-rexalation (hr)	1.01	34.5	35.5
Best Solution (hr)	36.75	36.75	36.75
CPU Time (s)	7,200	34.98	1.662
Optimality Gap (%)	0.68	0	0
Nodes	10,628	4,188	75

is given as $TC_1 = 36$ hours. This is calculated by using the parameter Γ. The maximum planned start time of the second plant wide cleaning task is calculated as $TC_1 + \Gamma + \alpha_1$ which is given as $TC_2 = 75$. Two plant wide cleaning tasks corresponds to the minimum number of plant wide cleans required for the horizon length. All data specific to industrial case study 1 are given in Appendix C.

In Figure 4.2 the results of the optimized production schedule (bottom) are compared against the historical production schedule (top). The historical schedule sequence, w.r.t product families, is observed as F8-F9 (PWC) F16 (PWC) F23-F14-F28 (PWC) F1-F2-F15-F26 (PWC) F20. The historical schedule has a total of 18.25 hours of changeover time, attributed to thirteen 0.25 hour changeovers, three 1 hour changeovers and four plant wide cleaning tasks. It is observed that the production sequence is completely rearranged in the optimized schedule and is given as F2-F16-F23 (PWC) F1-F26-F15 (PWC) F20-F8-F9-F14-F28. The optimized schedule has a total of 13.5 hours of changeover time, attributed to fourteen 0.25 hour changeovers, four 1 hour changeovers and 2 plant wide cleans. The historical schedule has a makespan of 107.25 hours while the optimized production schedule has a makespan of 102.5 hours. The reduction in the scheduled makespan is attributed to the removal of two plant wide cleaning tasks, being replaced with one 0.25 hour changeover and a 1 hour changeover. This reduction of 4.75 hours of changeover time corresponds to a

Figure 4.2: Industrial Case Study 1 - The historical schedule is given as the top Gantt chart and the optimized schedule is given as the bottom.



35% reduction in the downtime of the target facility for this production week. Also it can be seen the historical makespan is within 5% of the proven optimum.

The computational performance of models M_D, M_C and M_S is given in Table 4.2. Symmetry aggregate model M_S is observed to have the best computational performance. Model M_S proves the makespan of 102.5 hours is the global optimum in 5,834 CPUs, while model M_C fails to find the optimal solution after 43,200 CPUs and is left with an optimality gap of 8.65%. In addition, model M_S is observed to find the optimum solution in 309 CPUs, as opposed to model M_C which finds its best solution in 15,000 CPUs. Model M_D found 1 feasible solution of 106.75 hr makespan at 41,500 CPUs. After 43,200 CPUs model M_D was left with an optimality gap of 30.14%. It is noted the discrete time model schedules 4 process wide cleaning tasks in comparison to the the 2 deemed optimal by the continuous time models.

Table 4.2: Industrial case study 1 - computation performance

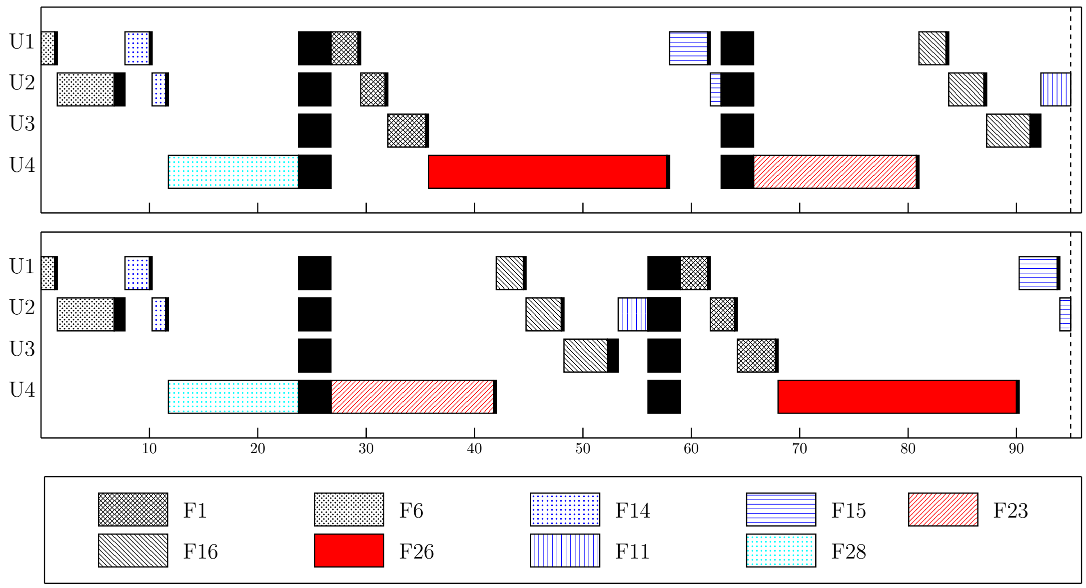
	M_D (Δt=0.25hr)	M_C	M_S
Event Points	432	24	14
Constraints	144,323	94,326	43,517
Binary Variables	9,482	1,106	1,010
Continuous Variables	102,906	111,097	33,699
LP-relaxation (hr)	0.2359	95	97.5
Best Solution (hr)	107	104	102.5
Time to Best Solution (s)	41,500	15,000	309
CPU Time (s)	43,200	43,200	5,834
Optimality Gap (%)	30.14	8.65	0
Nodes	5,091	22,106	236,577

4.3 Scheduling Case Study 2

The second industrial case study is derived from historical production week “Z” and requires 16 production targets to be met within 98 hours ($H = 98$). The 16 tasks belong to 9 product families. Product families F15 and F26 belong to production group 3 and families F14 and F28 belong to production group 2. Two plant wide cleaning tasks (PWC) are scheduled to occur within the production schedule, with $TC_1 = 36$ and $TC_2 = 75$. All data unique to case study 2 are reported in Appendix C.

A comparative Gantt chart of the historical production schedule (top) and the optimized production schedule (bottom) is given in Figure 4.3. The optimized and historic schedule are observed to have a makespan of 95 hours. The product sequence of the historical schedule is observed as F6-F14-F28 (PWC) F1-F26-F15 (PWC) F23-F16-F11, while the sequence of the optimized schedule is observed as F6-F14-F28 (PWC) F23-F16-F11 (PWC) F1-F26-F15 Although the optimized and historical schedule are different they are observed as equivalent. This is because the sequencing of product families in between the PWC tasks are identical. It is important to note that in any schedule the groups of product families

Figure 4.3: Industrial case study 2 – The historical schedule is given as the top Gantt chart and the optimized schedule is given as the bottom.



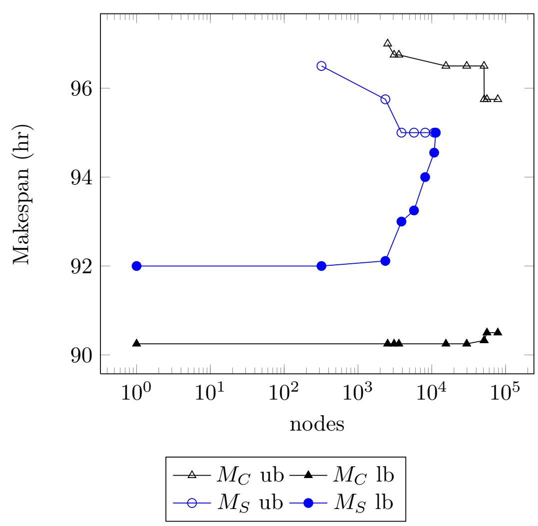
between plant wide cleaning tasks can be permuted in the overall sequence without effecting the makespan. Both schedules have a total of 11.25 hours of changeover time, attributed to eleven 0.25 hour changeovers, two 1 hour changeovers and 2 plant wide cleanings.

The computational performance of models M_D, M_C and M_S is listed in Table 4.3. Symmetry aggregate model M_S is observed to have the best computational performance, proving the makespan of 95 hours is the global optimum in 190 CPUs. Model M_C fails to find the optimal solution after 43,200 CPUs and is left with an optimality gap of 5.48%. Model M_D fails to find a feasible solution after exploring 213 nodes in 43,200 CPUs. The progression of the upper bound (ub) and lower bound (lb) of models M_C and M_S is plotted against nodes searched in Figure 4.4. It can be observed that the upper and lower bounds of model M_S begin and remain within the upper and lower bound of model M_C. From this we can infer that model M_S provides a tighter relaxation to the problem in comparison to model M_C.

Table 4.3: Industrial case study 2 - computation performance

	M_D (Δt=0.25hr)	M_C	M_S
Event Points	392	19	12
Constraints	129,628	67,681	36,233
Binary Variables	6,681	686	698
Continuous Variables	102,181	88,142	29053
LP-rexalation (hr)	2.27	90.25	92
Best Solution (hr)	–	95.75	95
Time to Best Solution (s)	–	27,000	80
CPU Time (s)	43200	43,200	190
Optimality Gap (%)	–	5.483	0
Nodes	213	43,212	11,320

Figure 4.4: Optimization upper bound (ub) and lower bound (lb) progression of models M_C and M_S for case study 2



4.4 Scheduling Case Study 3

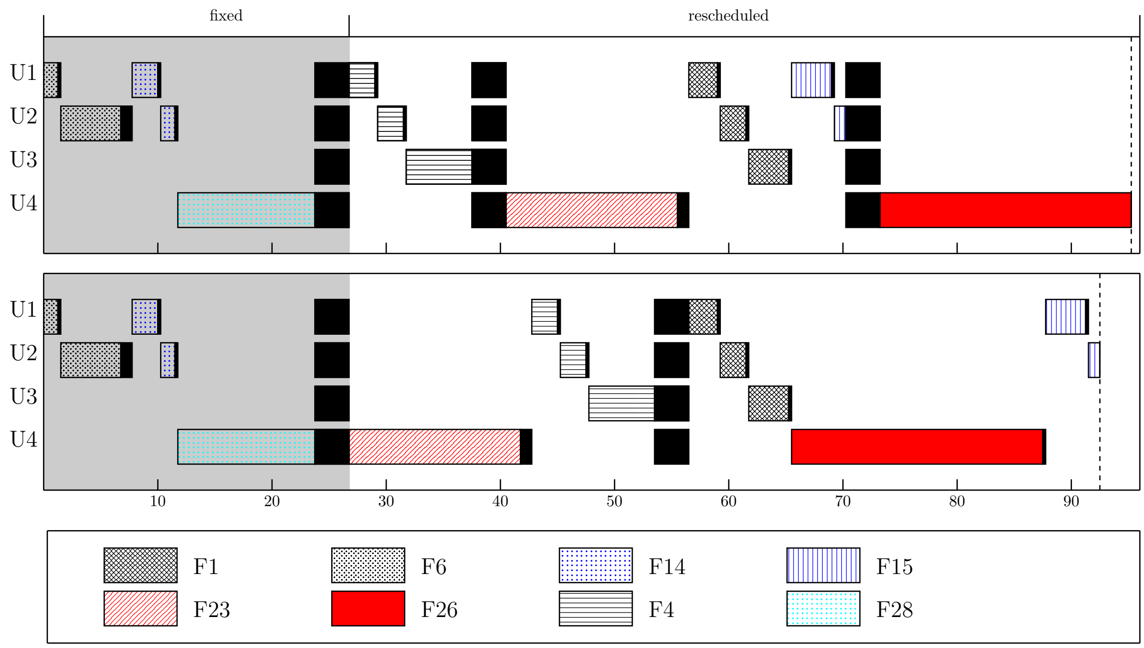
The third case case is derived from a process reschedule of production week "Z". A reschedule was required as a shipment of 5.91 weight equivalents of raw material Rm35 was received a week early and was required to be used. In addition, raw material Rm46 would not be available for processing until 56 hours into the processing scheduling. The plant's response was to modify the production targets such that the shipment of Rm35 would be used within the production week; modified production targets are listed in Appendix C. Product family F4 belongs to production group 1. To mimic the rescheduling information available to the plant management team, the first day of the historical production schedule was fixed in the rescheduled optimization. The unavailability of raw material Rm46 is included in the optimization via the material receipt methodology originally proposed by Maravelias and Grossmann [2003a] (Appendix B). The timing of the material receipt is specified as $TK_1 = 56$ hours, the amount of the delivery is given as $AD_1 = 8.44$ and this delivery belongs to set L_{Rm46}.

Figure 4.5 depicts the Gantt chart of the rescheduled historical production schedule (top) and the optimized rescheduled production schedule (bottom). It is observed that the third process wide cleaning task, included by plant employees, was not necessary to complete the production schedule.

The historical production reschedule has a makespan of 95.25 hours while the optimized reschedule has a makespan of 92.5 hours. The saving of 2.75 hours is gained by rearrangement of the production sequence such that the third process wide clean is removed and replaced with a 15 minute changeover. This results in a 21% reduction in the total changeover time and proves the historical reschedule lies within 3% of the plant optimum.

The computational performance of models M_C and M_S are listed in Table 4.4. It is noted model M_D was excluded from the optimization as it had previously been incapable of solving the planned production schedule. Symmetry aggregate model M_S is observed to have the best computational performance; the model proves the makespan of 92.5 hours is the global optimum in 2.54 CPUs. Model M_C also finds the optimum solution and proves optimality in 228 CPUs and 14,016 nodes. These results are interesting as both

Figure 4.5: Industrial case study 3 – The historical reschedule is given as the top Gantt chart and the optimized reschedule is given as the bottom.



models exhibit significant performance improvements in this case study. Although fixing the first portion of the production schedule reduces problem size, it alone can not explain the large improvement in computational performance. It is hypothesized that the additional processing restrictions aid in reducing the number of symmetrical solutions.

4.5 Scheduling Case Study 4

The final scheduling case study is derived from historical production week"Y" and requires 18 production targets to be met within a scheduling horizon of 135 hours ($H = 135$). The 18 processing tasks can be grouped into 12 product families, with families F4 and F24 belonging to production group 1 and families F25 and F3 belonging to production group 4.

Three plant wide cleaning tasks are specified to occur within the scheduling horizon, with $TC_1 = 36$, $TC_2 = 75$, and $TC_3 = 114$. Three plant wide cleans is the minimum number required given the horizon length. All data specific to industrial case study 4 is given in

Table 4.4: Industrial case study 3 - computation performance

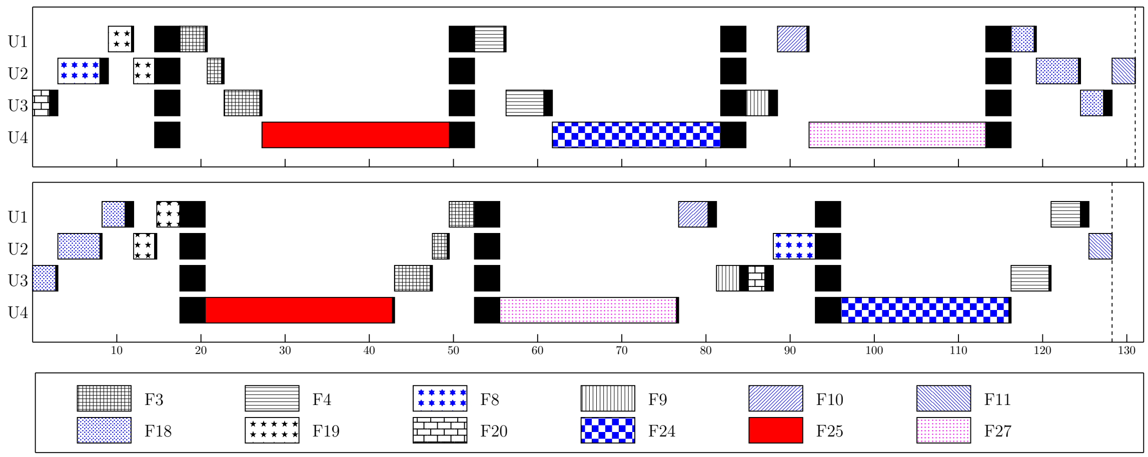
	M_C	M_S
Event Points	18	11
Constraints	66,637	32,795
Binary Variables	662	592
Continuous Variables	88,261	26,680
LP-rexalation (hr)	89.75	91
Best Solution (hr)	92.5	92.5
Time to Best Solution (s)	30	0.5
CPU Time (s)	228	2.54
Optimality Gap (%)	0	0
Nodes	14,016	81

Appendix C.

In Figure 4.6 the results of the optimized production schedule (top) are compared against the historical production schedule (bottom). The historical schedule has a makespan of 131 hours while the optimized production schedule has a makespan of 128.25 hours. The historical schedule has a total of 19 hours of changeover transitions, comprised of eight 0.25 hour changeovers, five 1 hour changeovers and 4 plant wide cleans. The optimized schedule has a total of 16.25 hours of changeover transitions, this reduction in changeover time is accomplished by replacing 1 plant wide clean with a 0.25 hour changeover. This removal 2.75 hours in process cleaning corresponds to a 14.5% reduction in the cleaning time of the target facility. It can also be seen the historical schedules makespan is within 2% of the best solution.

The computational performance of models M_C and M_S is given in Table 4.5. Symmetry aggregate model M_S is observed to have the best computational performance. Model M_S finds a solution of 128.25 hour makespan in 400 CPUs and is left with a optimality gap of 4.13% after exploring 189,894 nodes in 7,200 CPUs. Model M_C finds a solution of 129 hour makespan in 3,700 CPUs and is left with a optimality gap of 6.21% after exploring

Figure 4.6: Industrial Case Study 4 - The historical schedule is given as the top Gantt chart and the optimized schedule is given as the bottom.



44,604 nodes in 43,200 CPUs. Model M_D was not used in the case study evaluation due to poor performance on smaller sized problems previously showcased. In this case study neither model is able to prove optimality of the provided solutions. The case study requires 21 and 16 event points for models M_C and M_S respectively and represent the largest case study with respect to model size. The performance degradation exhibited is consistent with the observations of Méndez *et al.* [2006] who stated that 15 event points appears to be the upper limit of model complexity for global event point continuous time models.

4.6 Computer Aided Policy Evaluation

The case studies presented above help aid in highlighting the potential efficiency gains possible using optimization to schedule production within a consumer foods production facility. This is however not the only application such optimization models can be used for; it is also possible to use such models to evaluate what-if type scenarios. For example, one may modify the production rates in anticipation of purchasing new packaging equipment. The optimized schedules could then be used as a quantitative analysis to ascertain the best possible improvements or even highlight why production gains may not meet expectations. Such an opportunity presented itself during the term of this research, as the production facility planned to undergo a retrofit to increase production capacity. New equipment in

Table 4.5: Industrial case study 4 - computation performance

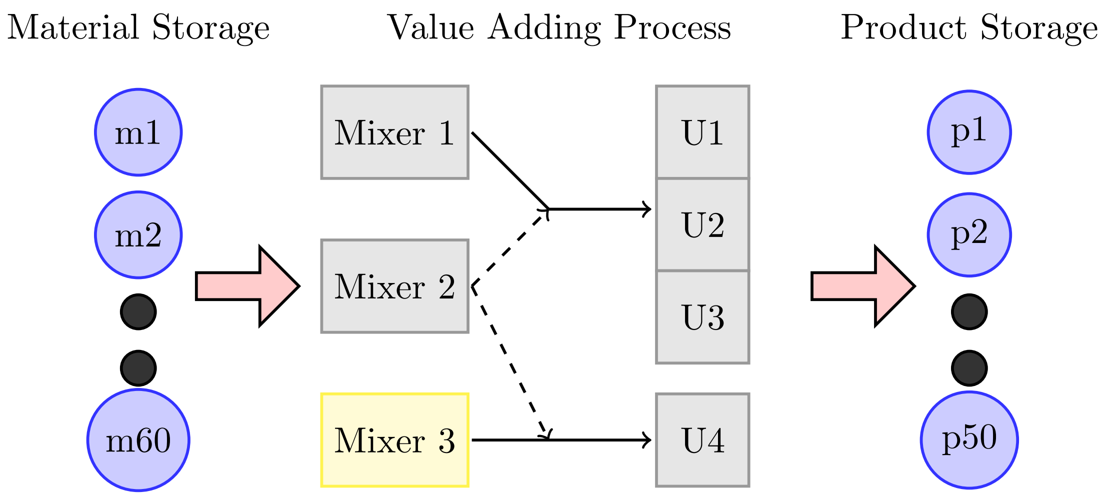
	M_C	M_S
Event Points	21	16
Constraints	82,704	51,798
Binary Variables	948	1,170
Continuous Variables	104,589	39,681
LP-rexalation (hr)	121	122.5
Best Solution (hr)	129	128.25
Time to Best Solution (s)	3,700	400
CPU Time (s)	43200	7,200
Optimality Gap (%)	6.21	4.13
Nodes	44,604	189,894

the mixing train and packaging lines would allow for packaged (U1,U2,U3) and specialized (U4) packaging lines to run in parallel.

Mixer one acts as a dedicated feed to packaged production line (units 1-3) and mixer 3 is dedicated to specialized production line (unit 4), as seen in Figure 4.7. Mixer 2 now acts as a swing mixer to feed either packaging line. Two different production strategies were designed by plant management to take advantage of this new production flexibility. It is was desired to evaluate each strategy to determine which would provide the best improvements in plant throughput. The aggregate model M_S was chosen to perform such evaluations due to its superior computational performance.

Strategy 1 In strategy 1 (S1) mixer 2 is used to swing mixing capacity between the production lines. When the specialized production line is being utilized mixer 2 will feed unit 3, and only mixer 1 will feed the packaged production line. In this mode only packaged products similar to the current specialized product are made. When specialized production is not required mixer 2 will be used to feed the packaged production line, in tandem with mixer 1. The set of similar product families is represented by $F_{f'}$ and is defined as the prod-

Figure 4.7: Description of plant post expected retrofit



uct family f that is deemed similar to product family f'. In this representation the notation f will denote packaged products while f' denotes specialized products. This requirement can be stated that if a specialized product family f' is starting or actively processing so must the respective packaged product $f \in F_{f'}$.

$$Ws_{fn} + \sum_{n'<n} Ws_{fn'} - \sum_{n'\leq n} Wf_{fn'} = Ws_{f'n} + \sum_{n'<n} Ws_{f'n'} - \sum_{n'\leq n} Wf_{f'n'} \quad \forall f',\ f \in F_{f'},\ n \tag{4.1}$$

Constraint (4.1) ensures that product families f' and $f \in F_{f'}$ operate simultaneously in the production schedule. The above constraint allows for product families f' and $f \in F_{f'}$ to finish at or before the same event point. This allows for some idle time on either unit if it is determined optimal. Constraint (4.1) is added to model M_S and is defined as model M_{S1}.

Strategy 2 Strategy 2 (S2) involves continuous parallel operation of the packaged and specialized production lines with mixer 2 dedicated to specialized production at all times. In this strategy any packaged product may be manufactured while the specialized production line is in operation; in accordance to previously specified operational policies. The plant operators highlighted that it would be preferable to run similar product families at similar times to provide simplified operations. As such the constraint given above can be restated as:

$$Ws_{fn} + \sum_{n'<n} Ws_{fn'} - \sum_{n'\leq n} Wf_{fn'} \leq Ws_{f'n} + \sum_{n'<n} Ws_{f'n'} - \sum_{n'\leq n} Wf_{f'n'} \quad \forall f',\ f \in F_{f'},\ n \tag{4.2}$$

Constraint (4.2) enforces that specialized product families f' must be operating if the respective product family $f \in F_{f'}$ is also operating. However the constraint allows for specialized product families to start or continue processing even if product family $f \in F_{f'}$ is not operating. This allows for additional scheduling flexibility when compared to operational strategy 1. Model M_S is augmented with constraint (4.2) and referenced as model M_{S2}.

4.6.1 Case Study 1 Revisited

The goal of this case study is to evaluate the proposed operational strategies and determine the expected gains in plant throughput. Historical data from case study 1 will be used as a basis; the expected production rates for each strategy are listed in Appendix C. The results of each production strategy will be compared against the optimal solution of case study 1 and the time horizon of interest is set to $H = 102.5$ hours, the minimum makespan of case study 1.

The results for throughput optimization for models M_{S1} and M_{S2} are given as the second and third Gantt chart in Figure 4.8, respectively. The optimized schedule in case study 1 (first Gantt chart) corresponds to a throughput of 196.83 weight equivalents of product. Operational strategy 1 and strategy 2 yield a throughput of 293.88 and 324.17 weight equivalents (we) of product, respectively. This corresponds to a increase of 49.3% throughput for strategy 1 and 64.7% increase in plant throughput for strategy 2. The increase in throughput of S2 over S1 is attributed to an increase in the amount of specialized product produced. The ability to run the packaged production line in parallel, albeit at slower than historic processing rates, allows for packaged production requirements to be met as well. It can therefore be concluded that operational strategy 2 allows for the greatest improvements in plant throughput.

Figure 4.8: Industrial case study 1 revisited with results of operational strategies 1 (middle) and 2 (bottom).

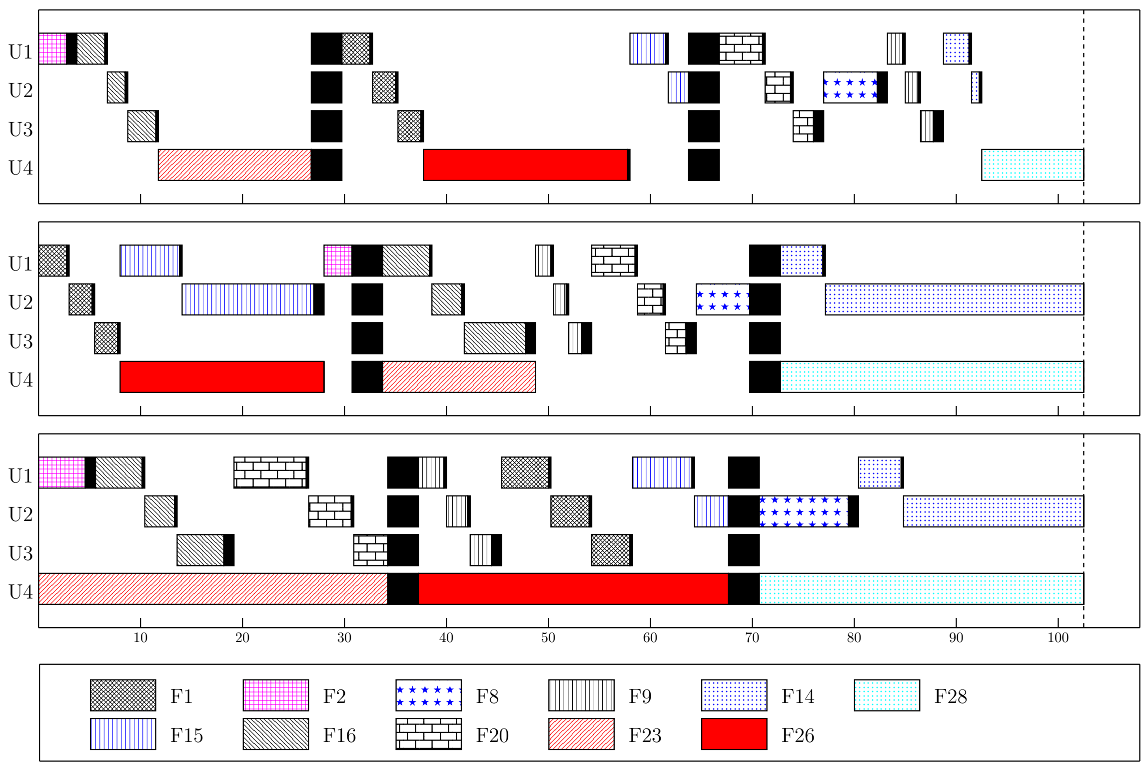


Table 4.6: Policy evaluation case study - computational performance, objective given in weight equivalents (we)

	M_{S1}	M_{S2}
Event Points	11	11
Constraints	33,221	33,221
Binary Variables	788	788
Continuous Variables	23,623	23,623
LP-rexalation (we)	397.17	331.73
Best Solution (we)	293.88	324.17
Time to Best Solution (CPUs)	70	115
CPU Time (s)	1461	881
Optimality Gap (%)	0	0
Nodes	117,342	17,394

The computational performance of models M_{S1} and M_{S2} are given in Table 4.6. An important note is the difference in computational time required to prove optimality when one compares the two objective functions. Throughput maximization requires far less computation time to prove optimality of solution, acheiveing reductions of 4,953 CPUs and 219,183 nodes searched for operational strategy 2, and 4,373 CPUs and 59,235 nodes searched for operational strategy 1.

Chapter 5

Production Planning

The focus of the chapter is to expand from the short-term scheduling optimization to medium-term production planning optimization. Production planning optimization is conducted over a medium-term horizon to determine the most efficient production targets and inventory levels of the industrial supply chain. To accomplish this task, a planning optimization model must be developed to account for the unique characteristics and policies of the target facility. As the production facility exhibits sequence dependent production changeovers the true operating capacity is variable in nature. Failure to account of this in a planning model can lead to capacity overestimation and the generation of weekly production targets which can not be met at the detailed scheduling level. This chapter will present the development of a planning optimization model which uses traveling salesman sequencing constraints to provide an accurate bound on the true capacity of the production plant. Such use of traveling salesman sequencing constraints has been presented previously by Erdirik-Dogan and Grossmann [2007], in which the authors develop a planning model that provides accurate planning solutions over medium-term horizon lengths.

The planning model will then be used to investigate the potential benefits of planning optimization at the production plant through examination of several case studies. The first case study will apply the planning model to a three month planning horizon with one week planning intervals. The case study will also examine the effect of external minimum production targets specified for 6 key product SKUs. The second case study will investigate the potential benefit of changing the time limit between the process wide cleaning events (Γ). This will be investigated with and without key product minimum production targets.

5.1 Planning Model Formulation

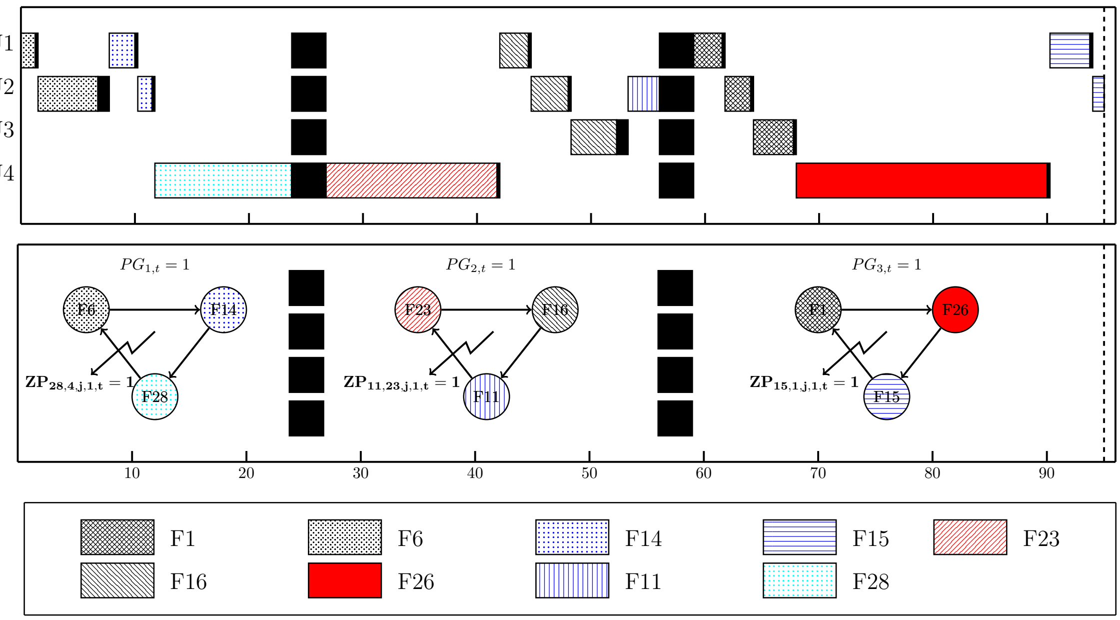
To provide accurate bounds on plant capacity the planning model must account for the production grouping rules, the cleaning policies of the target facility and the sequence dependent changeovers between production tasks. The planning model presented is based on the scheduling rules presented in Chapter 3, such that key policies are enforced but detailed task timing is avoided.

In the production planning formulation the index t represents the planning intervals, which are defined as one week in length. Let index i represent processing tasks, index j represent production equipment and index g represent production groups. The sets I_j and I_g represent the set of tasks that can be run on unit j and the set of tasks that can be run in production group g, respectively. Index f is used to represent product families and the sets F_j and F_g represent the product families that can be run on equipment units j and in productions groups g, respectively. Similar to the aggregate scheduling formulation, processing tasks $i \in I_f$ are grouped together into product families and sequencing is carried out between product families.

In the planning model, the active production groups in each planning interval t are selected via binary indicator PG_{gt}. This is done for several reasons: First, the production groups represent sequences of product families that share similar product formulations and must be run in-between plant wide cleans. Therefore, by grouping production into production groups g product family sequences can be developed via traveling salesman based constraints. Second, if the number of active production groups is known it is possible to determine the time required for conducting plant wide cleaning tasks.

Figure 5.1 depicts this planning formulation in relation to a typical process schedule, displaying the grouping based on product family and production groups. The product families are seen as tasks with identical fill patterns and colors, whereas production groups are the sequences of tasks between plant wide cleans. The planning formulation deconstructs this

Figure 5.1: Construction of cyclical production schedules using traveling salesman based sequencing constraints



process schedule into three production groups, which consist of product families: PG_1 - F6, F14, F28; PG_2 - F23, F16, F11; and PG_3 - F1, F26, F15. The changeover time required for each production group sequence is determined through the use of travelling salesman problem (TSP) type constraints. In each production group g the active product families f are placed in cyclical sequences, with links representing the changeover time required to switch production between product families. The planning formulation then breaks the cyclical sequence at the link with the largest changeover duration to produce a linear production sequence. Plant wide cleaning tasks are accounted for through the restriction of the available processing time in each planning interval t.

Batch Size and Processing Time Constraints

The batch sizing of task i is accomplished through variable FP_{ijgt}, which represents the batch size of processing task i in production group g on unit j in planning period t. In addition, binary variable YYP_{ijgt} is introduced to indicate if task i in production group g on unit j is active in planning period t. The following constraints are imposed:

$$B_i^{min} YYP_{ijgt} \leq FP_{ijgt} \leq B_i^{max} YYP_{ijgt} \qquad \forall j,\ g,\ i \in I_j \cap I_g,\ t \tag{5.1}$$

Constraint (5.1) enforces the batch size of task i on unit j in production group g at planning interval t to lie within the batch size limits, as given by parameters B_i^{min} and B_i^{max}. The total processing time of a batch of task i is tracked through PT_{ijgt} and is related to the batch size of task i through the following constraints:

$$PT_{ijgt} = \alpha_i YYP_{ijgt} + \beta_i FP_{ijgt} \qquad \forall i \in I_j \cap I_g,\ j,\ g,\ t \tag{5.2}$$

Constraint (5.2) defines the total processing time of task i to be a combination of the fixed (α_i) and variable (β_i) processing rates. α_i and β_i remain consistent with the parameters defined for the scheduling formulation.

Material Balance

To track the transformation of material states throughout the planning horizon, variable S_{st} is introduced as the ending inventory of state s in planning interval t. Production is carried out according to a production recipe, which stipulates the relative proportion of material state input (ρ_{si}) and the proportion of material state output ($\bar{\rho}_{si}$). Let K_i and P_i represent the set of units j capable of running task i and the set of production groups g that include task i, respectively. The following constraint is enforced:

$$\begin{aligned} S_{st} = S_{s,t-1} &+ \sum_{i \in O_s} \sum_{j \in K_i} \sum_{g \in P_i} \bar{\rho}_{si} FP_{ijg,t} \\ -\sum_{i \in I_s} \sum_{j \in K_i} \sum_{g \in P_i} \rho_{si} FP_{ijgt} &- SS_{st} + MO_{st} \qquad \forall s, t \end{aligned} \tag{5.3}$$

Constraint (5.3) simply states that the ending inventory of state s in planning period t is given as the balance on the previous inventory and the amount of state s produced, consumed, sold or purchased in planning interval t. SS_{st} is defined as the amount of state

s sold to market in planning interval t, while MO_{st} defines the amount of material state s purchased. $\bar{\rho}_{si}$ and ρ_{si} remain consistent with the parameters defined in the scheduling formulation.

Inventory Limit Constraints

The inventory level of product and material states is often bounded by stocking policy and warehouse limitations.

$$C_s^{min} \leq S_{st} \leq C_s^{max} \qquad \forall s,\ t \tag{5.4}$$

Constraint (5.4) enforces the current inventory level of state s to be bounded between minimum (C_s^{min}) and maximum (C_s^{max}) limits.

Binary Variable Restrictions

In addition to the above constraints the following binary variable restrictions are imposed. As tasks are grouped into product families no task should be selected unless the appropriate product family is also selected. YP_{fjgt} is a binary variable that is equal to 1 if product family f is selected to be run on unit j in production group g at planning period t.

$$YYP_{ijgt} \leq YP_{fjgt} \qquad \forall i \in I_f,\ f \in F_g \cap F_j,\ j,\ g,\ t \tag{5.5}$$

The activity of processing tasks i is restricted to coincide with with the occurrence of the appropriate product family f, as given by constraint (5.5). A similar restriction is imposed for the selection of product families, as these families must be run within a applicable production group. To accomplish this binary variable PG_{gt} is introduced and equated to 1 if production group g is active in planning period t, and zero other wise.

$$YP_{fjgt} \leq PG_{gt} \qquad \forall f \in F_g \cap F_j,\ j,\ g,\ t \tag{5.6}$$

Constraint (5.6) restricts binary indicator variable YP_{fjgt} to occur only if the appropriate production group g is active at planning period t.

Demand Satisfaction

Two alternative ways to address demand satisfaction are addressed within the proposed planning model.

$$SS_{st} = D_{st} - BL_{st} \qquad \forall s,\ t \tag{5.7a}$$

$$SS_{st} \geq D_{st} \qquad \forall s,\ t \tag{5.7b}$$

Constraint (5.7a) enforces that the sale of any product state must be equal to the demand minus any incurred backlog. The allowable backlog for any product state s at planning interval t is limited to at most 5% of the demand at that planning interval ($BL_{st} < 0.05D_{st}$). Constraint (5.7b) allows for sales to greater than the specified demands and does not allow for any backlog. It is important to track the accumulated backlog of states throughout the planning horizon. To accomplish this variable TBL_{st} is introduced and defined as the total backlog of state s at planning interval t.

$$TBL_{st} = TBL_{s,t-1} + BL_{st} \qquad \forall s,\ t \tag{5.8}$$

Constraint (5.8) tracks the accumulation of backlog of state s throughout the planning horizon. To ensure all demands are met the total backlog for any state s is required to be zero at the end of the planning horizon ($TBL_{s,t=|T|} = 0 \forall s$).

Production Targets

In some plants, production targets for key products are often assigned to be met within planning periods. This is usually done to ensure uninterrupted product supply or is developed in conjunction with key strategic partners over a longer term horizon.

$$\sum_{i \in O_s} \sum_{j \in K_i} \sum_{g \in P_i} \bar{\rho}_{si} FP_{ijgt} \geq Prod_{st} \qquad \forall s,\ t \tag{5.9}$$

Constraint (5.9) defines the minimum amount of state s that must be produced in planning interval t to meet specified production targets.

Sequencing Constraints

Sequencing of active product families is accomplished through binary variable $ZP_{f,f',j,g,t}$, which is 1 if product family f' is sequenced after product family f on unit j in product group g in planning period t. Breaking one of the links in each active production group leads to the generation of an applicable product family sequence for that group. $ZZP_{f,f',j,g,t}$ is introduced and defined as 1 if the link between product family f and f' on unit j in production group g in planning period t is broken. The optimal sequence is determined by breaking the cyclical schedules at link $ZP_{f,f',j,g,t}$. The following sequencing constraints are imposed:

$$YP_{f,j,g,t} = \sum_{f' \in F_g} ZP_{f,f',j,g,t} \qquad \forall j,\ g,\ f \in F_g \cap F_j,\ t \tag{5.10}$$

$$YP_{f',j,g,t} = \sum_{f \in F_g} ZP_{f,f',j,g,t} \qquad \forall j,\ g,\ f' \in F_g \cap F_j,\ t \tag{5.11}$$

Constraints (5.10) and (5.11) state that product family f can only be included in planning period t if product family f participates in both a forward and backward link to other product families. The above constraints will generate a cyclical schedule for each production group g active in planning period t. To determine the optimal sequence the cycle is broken at the link with the highest changeover time.

$$\sum_{f \in F_j \cap F_g} \sum_{f' \in F_j \cap F_g} ZZP_{f,f',j,g,t} = PG_{g,t} \qquad \forall j,\ g,\ t \tag{5.12}$$

Constraint (5.12) enforces that one link is broken in each active production group. If a production group is not active, sequencing variables $ZP_{f,f',j,g,t}$ will be forced to zero through constraints (5.6), (5.10) and (5.11) and $ZZP_{f,f',j,g,t}$ will equate to zero through the following restrictions.

$$ZZP_{f,f',j,g,t} \leq ZP_{f,f',j,g,t} \qquad \forall f, f' \in F_j \cap F_g,\ j,\ g,\ t \tag{5.13}$$

To ensure that only valid links are broken, constraint (5.13) enforces that only links between family f and f' that exist can be broken. If no sub-cycles exist then the above sequences will correspond to schedules with sequence dependent changeovers taken explicitly into account. If sub-cycles exist then the sequences generated will correspond to a valid lower bound on the changeover times, and the model will generate a valid upper bound on the achievable profit.

Self-loop Restrictions

A prominent issue when using traveling salesman based sequencing constraints is the appearance of self-loops within the optimization solutions. Self-loops are the occurrence of a product family with sequencing connections to itself, which will cause a underestimation in the amount of changeover time. It is therefore necessary to restrict the occurrence of self-loops such that they only occur if intended.

$$YP_{fjgt} \geq ZP_{ffjgt} \qquad \forall j,\ g,\ f \in F_j \cap F_g,\ t \tag{5.14}$$

$$ZP_{ffjgt} + YP_{f'jgt} \leq 1 \qquad \forall j,\ g,\ f, f' \in F_g \cap F_j,\ t \mid f \neq f' \tag{5.15}$$

$$ZP_{ffjgt} \geq YP_{fjgt} - \sum_{\substack{f' \in F_g \cap F_j \\ f' \neq f}} YP_{f'jgt} \qquad \forall j,\ g,\ f \in F_g \cap F_j,\ t \tag{5.16}$$

If product family f is the only active product family in production group g on unit j in planning period t then constraint (5.16) ensures that binary indicator variable ZP_{ffjgt} is equal to 1. If any other product family $f' \neq f$ is also active in production group g on unit j in planning period t then constraint (5.15) restricts binary indicator variable ZP_{ffjgt} to equal 0. The above constraints are effective in preventing unnecessary self-loops from existing in the optimized production plans.

Changeover Times

In the above aggregate formulation, two types of changeovers exist: transitions between tasks $i \in I_f$ and transitions between sequenced product families. To account for the transition times variables $ICTV_{fjgt}$ and $TRNP_{jgt}$ are introduced and defined as follows: $ICTV_{fjgt}$ is the total changeover time between production tasks $i \in I_f$, while $TRNP_{jgt}$ is the total changeover time between product families active on unit j in production group g in planning period t. The following constraints are imposed:

$$ICTV_{fjgt} = \gamma_f \left(\sum_{i \in I_f} YYP_{ijgt} - YP_{fjgt} \right) \qquad \forall f \in F_g \cap F_j,\ j,\ g,\ t \tag{5.17}$$

Constraint (5.17) determines the changeover time required within product family f. The number of required changeovers is equal to the number of active tasks minus 1. γ_f is given as the changeover time required between tasks $i \in I_f$.

$$TRNP_{jgt} = \sum_{f,f' \in F_g \cap F_j} \left(sl_{ff'} ZP_{ff'jgt} - sl_{ff'} ZZP_{ff'jgt} \right) \qquad \forall j,\ g,\ t \tag{5.18}$$

Constraint (5.18) enforces that for every active link between product families the corresponding amount of changeover time is added to $TRNP_{jgt}$. $sl_{ff'}$ is given as the changeover time required to transition from family f to f'.

Production Group Time Limits

It is necessary to restrict the amount of time spent processing tasks in production groups to be less then 36 hours (Γ). To accomplish this, variable PGT_{jgt} is introduced and defined as the total processing time of production group g on unit j in planning period t. This includes the changeover time required between tasks in production group g. The following constraints are imposed:

$$\sum_{i \in I_g \cap I_j} PT_{ijgt} + \sum_{f \in F_j \cap F_g} ICTV_{fjgt} + TRNP_{j,g,t} = PGT_{jgt} \qquad \forall j,\ g,\ t \tag{5.19}$$

Constraint (5.19) enforces PGT_{jgt} to be the combination of task processing time, task changeovers and sequencing changeovers in production group g on unit j in planning interval t.

$$PGT_{jgt} \leq \Gamma \ PG_{gt} \qquad \forall j, g, t \tag{5.20}$$

Constraint (5.20) limits the total processing time of production group g on unit j to be less than Γ hours in planning interval t.

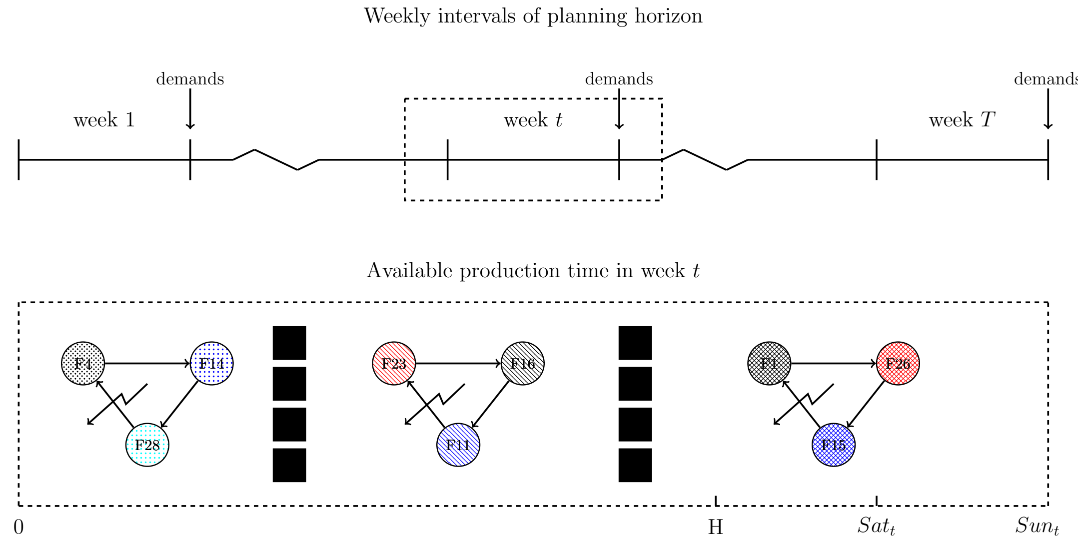
Horizon Time Balance and Limits

The production facility typically runs on a 5 day production schedule with the ability to schedule overtime if necessary. To include this into the above formulation, binary variables Sun_t and Sat_t are introduced and defined as: Sun_t is equal to 1 if Sunday overtime is scheduled for production period t, while Sat_t is defined as 1 if Saturday overtime is scheduled for production week t. The total available time for processing production groups then becomes variable in nature. A new variable HH_t is introduced to track the total available processing time in planning interval t. Figure 5.2 depicts the planning horizon time line and the available processing time in each interval. The following time balance is enforced:

$$HH_t = H - \tau \left(\sum_g PG_{gt} - 1 \right) + 24Sun_t + 24Sat_t \qquad \forall t \tag{5.21}$$

Constraint (5.21) is a balance on the total available production time of planning period t. The total time spent preforming process wide cleans can be determined as the number of active production groups minus one, and is represented in the second term on the R.H.S. τ is defined as the duration of a process wide cleaning operation. Saturday and Sunday

Figure 5.2: Representation of planning interval timeline and the processing time available in each interval



overtime both contribute 24 hours to the available production time in planning period t. H is defined as the base horizon time of the 5 day production week.

$$\sum_{g} PGT_{jgt} \leq HH_t \qquad \forall j, t \tag{5.22}$$

Constraint (5.22) then restricts the total processing time of all production groups g to be less than or equal to the total available production time of planning period t.

Objective

The objective of the optimization is the maximization of profit over the entire planning horizon, which is represented through variable Z. The profit of the planning horizon is defined as the revenue from demand satisfaction minus the associated production costs.

$$A^* = \sum_{t} \sum_{s \in Pro} \zeta_s SS_{st}$$

A^* represents the revenues of the planning period as given by the sale of product state s multiplied by the sales price of product state s (ζ_s). Pro is given as the set of product states. The major costs of the production facility are the material purchasing, inventory costs, the cost of production transitions and backlog penalties.

$$B^* = \sum_t \sum_{s \in Mat} \lambda_s MO_{st}$$

Material purchasing costs, B^*, is defined as the amount of material state s purchased at planning period t multiplied by the material costs (λ_s). Mat is defined as the set of raw material states. Inventory costs are divided into handling and storage cost for material states s.

$$C^* = \sum_t \sum_{s \in Pro} HC_s \sum_{i \in O_s} \sum_{j \in K_i} \sum_{g \in P_i} \bar{\rho}_{si} FP_{ijgt}$$

$$D^* = \sum_t \sum_{s \in Pro} SC_s S_{st}$$

Handling costs, C^*, are associated with the amount of product state s produced in each planning period t. HC_s is defined as the handling cost of one weight equivalent of product state s. Storage costs, D^*, are cost of storing product state s for each planning period t. SC_s is defined as the inventory storage cost of a weight equivalent of product state s over one planning period.

The transition costs are characterized by the total amount of time in changeover multiplied by the relative hourly transition cost.

$$E_1^* = \sum_j \sum_g \sum_t TC\ TRNP_{jgt}$$

$$E_2^* = \sum_j \sum_g \sum_t TC \sum_{f \in F_g \cap F_j} \gamma_f ICTV_{fjgt}$$

$$E_3^* = \sum_g TC\ \tau\ (\sum_t PG_{gt} - 1)$$

Table 5.1: Planning model constraint definitions

	Constraints	Objective
M_P	Constraints (5.1) to (5.6), (5.7b) and (5.10) to (5.22)	Constraint (5.23)
M_P^*	Constraints (5.1) to (5.6), (5.7a), (5.8) and (5.10) to (5.22)	Constraint (5.23)
M_P^T	Constraints (5.1) to (5.6), (5.7a) and (5.8) to (5.22)	Constraint (5.23)

The above defines the cost of transition for sequencing transitions, task transitions and process wide cleaning. TC is defined as the associated cost of one hour of transition time.

The objective function is then defined as:

$$Z = A^* - B^* - C^* - D^* - E_1^* - E_2^* - E_3^* \tag{5.23}$$

Equation (5.23) defines the overall profit of the planning horizon to be the balance between revenues and costs.

Planning Model Definitions

Three models are proposed from the above constraints: M_P - Planning model with demand satisfaction constraint (5.7b), M_P^* - Planning model with demand satisfaction constraint (5.7a), and M_P^T - Planning model with production target constraint (5.9) and demand satisfaction constraint (5.7a). The models are summarized in Table 5.1.

5.2 Planning Case Studies

This section will present two case studies that apply the models described above. The first case study investigates the application of model M_P^* to a 3 month planning horizon, using intervals of 1 week planning intervals. The additional sale of product material is not allowed and the solution should provide the lowest cost solution. The case study will also investigate the effect of additional production targets on the optimal solution of the model;

this will be done using model M_P^T. The second case study will investigate the effect of altering parameter Γ in a capacity constrained planning period. To accomplish this, model M_P is used to allow for sales over the specified demand. A optimality gap of 0% is set for all case studies and a time limit of 1 hour is specified. All models are implemented in GAMS 23.6.3 and solved with CPLEX 12.2.0.2 on a Intel® Core 2 Quad Q8200 at 2.33 GHz machine. Software was run in 64-bit Windows Vista Ultimate with 6 GBs of available RAM.

5.2.1 Planning Case Study 1

Models M_P^* and M_P^T are applied to a 12 week planning horizon for the target facility. The base horizon length is given as $H = 107$ hours and the production group time limit is given as $\Gamma = 36$ hours. The computational statistics for the 12 week planning period of models M_P and M_P^* is given in Table 5.2. Model M_P^* is considered the base solution and results for model M_P^T are scaled according the this base. LP-relaxations are scaled by division by the best solution provided by the model multiplied by 100. No sub-cycles are present in the final solutions and zero product backlogs are reported in either case. Both models fail to prove optimality but find very accurate solutions; within 0.5% of global optimum.

It is observed that model M_P^T provides a solution that is 88.62% that of the best solution provided by base model M_P^*. The addition of production targets thus negatively impacts the profitability of the optimal solution. It is observed that the reduction in profitability is due to a 17% increase in the overall costs. A summary of the operational operational costs of the solutions is given in Table 5.3. The overall costs are are scaled by dividing by the base solution and multiplying by 100. The individual contributions to the overall costs are also given.

The inventory profiles of the above solutions are provided in Figure 5.3. Grey columns represent inventory levels of model M_P^* while patterned columns represent the inventory levels of model M_P^T. It is clearly seen that the addition of the production targets markedly increases the inventory stock of the key product states, leading to increases in the associated costs of inventory, handling and material purchasing. It is also noted that model M_P^T required the scheduling of overtime to meet customer demands and the specified production

Table 5.2: Computational results of planning case study 1

	M_P^*	M_P^T
Constraints	69,793	69,865
Binary Variables	22,116	22,116
Continuous Variables	41,125	41,125
LP-relaxation (% Base)	100.75	101.4
Best Solution (Profit - % Base)	100	88.62
CPU Time (s)	3,600	3,600
Optimality Gap (%)	0.14	0.55
Nodes	4,758	7,454

targets. Four additional Sunday shifts and 3 additional Saturday shifts were scheduled within the horizon.

An interesting note is to compare the scheduled time in transition between the optimized production schedules against that of the historical plant operation. The average weekly time spent in transition is calculated as 10.88 hours for models M_P^* and 12.27 hours for model M_P^T. Review of historical production data over the same time horizon reveals the average weekly time spent in transition as 20.2 hours. As both models do not present sub-cycles it can be deduced that on average the weekly production transition time may be reduced by as much as 9 hours. This can be achieved through use of such a planning optimization system that exploits problem knowledge and leverages plant capabilities.

5.2.2 Planning Case Study 2

This case study will be used to investigate the effect of increasing the available time for processing production groups (Γ). According to operational policies, a production group is limited in total processing time to Γ = 36 hours. To better understand the effect this parameter has on plant operation, it was desired to investigate the efficiency gains possible as a result of increasing Γ. As such, several instances of Γ were run in 12 week planning

Figure 5.3: Optimized Inventory levels; products A) Ps34, B) Ps35, C) Ps36, D) Ps37, E) Ps38 and F) Ps39 (Model M_P^* solid bars; Model M_P^T patterned bars)

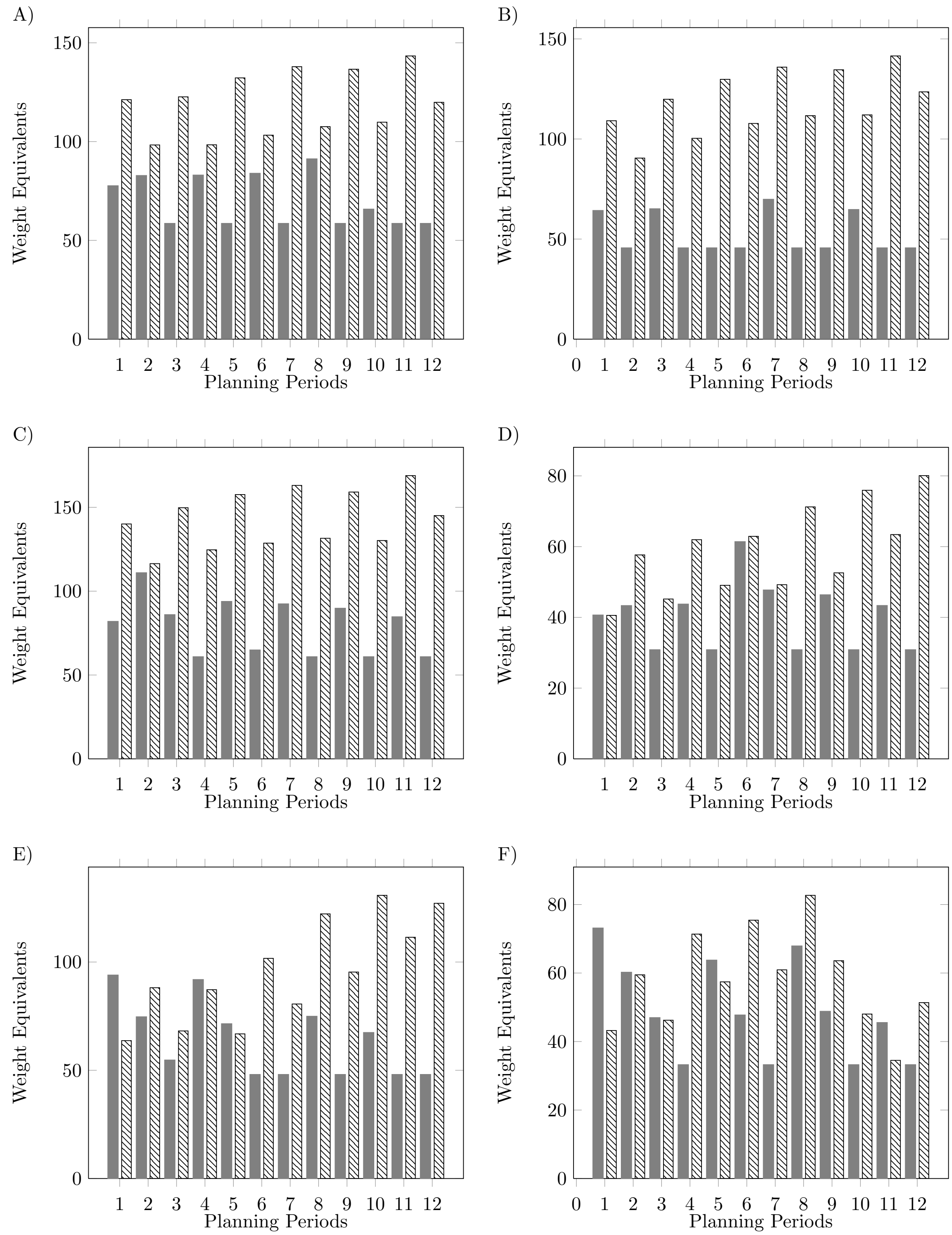


Table 5.3: Summary of operational costs of optimized production plan for planning case study 1. The percentage of costs is broken down as percentage of total costs.

	M_P^*	M_P^T
Costs	100	117.54
Materials	90.6	89.2
Handling	5.0	4.6
Inventory	3.1	3.6
Transitions	1.3	1.2
Over Time	0	1.4

horizons, which includes $\Gamma = 36$, 40, 44, 48 hours. Model M_P is used to allow for sales over the specified minimum demand. The upper bound on weekly state sales is bounded to be less than or equal to 200% that weeks demand.

Table 5.4 summarizes the optimal sales, profit and operational costs, as a percentage of the base case values, as Gamma is increased. Values are reported as percentage increase from the base, which is the solution value at $\Gamma = 36$. It is observed that the profitability of the plant increases marginally with increasing Γ. On average roughly 1% additional profit is achievable by increasing the length of time between process wide cleaning operations (Γ). Further investigation revealed that the gains in profitability are the result of decreases in the overall production changeover time. Table 5.5 displays a summary of the overall and individual types of process changeovers as Γ is increased. It is observed that the time spent performing process changeovers decreases (7.1, 9.7, 10.3 %) as Γ is increased. This decrease in overall changeover time is achieved by a reduction in the amount of plant wide cleaning operations (3 hr). As Γ is increased to 40, 44, and 48 hours the total time spent performing 3 hr changeovers is reduced to 114, 111 and 111 hours. This reduction in 3 hr changeover times is countered by an increase in the time spent performing 15 min and 1 hr changeovers compared to the $\Gamma = 36$ hr case. This makes sense as more products can be made in longer production groups and it is expected to see an increase in the shorter changeovers between similar product families.

Table 5.4: Comparison of sales, profits and costs when Γ is altered in planning case study 2, given as percentage increase over base value.

Γ	40	44	48
Sales	0.93	1.13	1.26
Profits	0.99	1.20	1.31
Costs	1.08	1.31	1.40

Table 5.5: Summary of total and type of process changeovers as Γ is increased for planning case study 2, given in hours.

Γ	36	40	44	48
Overall	199	185	179.75	178.5
15 min	19.75	23.75	23.5	23.5
1 hr	44.25	47.25	45.25	44
3 hr	135	114	111	111

It can therefore be deduced that Γ does not have a significant effect on the optimality of the production horizon over the range investigated. The increases yield less than 1.5% change in the profitability of the production plant. This is due to the fact that reductions in the number of require plant wide cleans is marked by an increase in shorter changeovers. It is determined that such initiatives to increase this limit, for the time range investigated, would yield only marginal increases in plant profitability.

Chapter 6

Conclusions and Recommendations

6.1 Chapter Conclusions

1. **Scheduling Model Formulation.** Three process scheduling optimization models are presented that are based on and extend the current discrete and continuous time modelling paradigms. Novel extensions to incorporate processing task grouping rules, based on product family and production group assignments, are incorporated into all three models, and reflect production policies currently practiced at the target facility. In addition, formulations to incorporate maintenance tasks with variable and dependent timing for continuous time models are presented. The formulation advances the field of continuous time modelling by allowing activities specified within the scheduling horizon to be dynamically assigned within the optimization procedure. The formulation opens a window to enhanced modelling of semi-continuous batch processes with complex production policies and maintenance activities. To overcome problem size and computational performance issues, an aggregate reformulation of the continuous time model is proposed. The model removes redundant solutions by ignoring task sequence, and instead is formulated to sequence product families. This removal of redundant solutions allows for model size reduction and improved computational performance.

2. **Scheduling Benefit Analysis.** Investigation into the potential benefit of scheduling

optimization to the processing facility in question was carried out through analysis of historical schedules to optimized schedules. It is shown that use of optimization at the scheduling level can result in a 5% improvement in scheduled makespan, and as such can improve plant capacity. Gains are made by rearrangement of processing tasks into more efficient production groups and reduce the number of scheduled process wide cleaning operations. A process reschedule is analyzed in a case study and similar improvements in production makespan are exhibited. In addition it is shown that due to model size and the requirement of fine discretization, discrete time models prove ineffective in providing a solution to this problem.

3. **Policy Evaluation.** During the course of the research the production facility underwent a plant retrofit to extend plant capacity and dissaggregate the production lines. It was desired to analyze the potential of two different modes of operation, as specified by plant management. The aggregate continuous time model is extended to reflect these two modes of operation and a case study is conducted to estimate the best operational performance capable. The scheduling model is successfully used to compare these modes of operation and highlight the best possible plant throughput.

4. **Production Planning.** Analysis of production scheduling case studies highlights the need to accurately account for sequence dependent changeovers, as changeovers represent approximately 15-20% of production time, according to historical operating data. A traveling salesman based approach is adopted and extended from literature to exploit the nature of the scheduling problem. The model accounts for sequence dependent changeovers, the required processing task grouping rules and plant wide cleaning policies. It is shown that a significant reduction in the time spent in changeover can be accomplished through improved inventory and production planning. In addition it is shown that specification of production targets for key product states can in some cases restrict the production of other products and reduce the effective capacity of the processing plant. It is shown that significant savings are achievable by lowering product inventory and reducing operational costs.

6.2 Recommendations for Further Work

1. **Production Planning Under Uncertainty** Production planning is typically carried out with forecasts of future demand patterns. Forecasts are routinely treated as deterministic and planning is conducted around uncertain future demands. In a fast moving consumer packaged goods industry, demands can vary within a statistical range and such information should be accounted for to plan production. Much opportunity exists to develop robust optimization approaches to incorporate the demand uncertainty information within production planning models. Exploitation and evaluation of such optimization models should be characterized to better understand the gains to industry for adopting such methodologies.

2. **Integration of Planning and Scheduling** Mismatch often appears between production planning and production scheduling models. The time spans and differences in detailed information lead to accuracy differences and possibly production targets that can not be implemented at the scheduling level. A method to efficiently and easily integrate the two levels is needed in industry to aid in planning execution and improve scheduling decisions.

Nomenclature

Indicies

n	Event points
s	States
j	Equipment units
i	Tasks
c	Maintenance events
g	Production groupings
f	Product families

Sets

I_j	The set of tasks i capable of being processed on unit j
I_c	Maintenance tasks i associated with maintenance event c
I_f	The set of tasks i associated with product family f
I_g	The set of tasks i in production group g
F_j	The set of product families f capable of production on unit j
F_g	The set of product families f in production group g
K_i	The set of units j that can perform task i
P_i	The set of production groups g that are associated with task i
SI_i	The set of all states s that are inputs to processing task i
SO_i	The set of all states s that are outputs of processing task i
I_s	The set of all processing tasks i that consume state s
O_s	The set of all processing tasks i that produce state s
ZW	The set of all tasks that produce at least one zero-wait state
Pro	The set of all product states
Mat	The set of all raw material states

Parameters

H	Time horizon length
H^*	Time horizon upper bound with slack allowance
α_i	The fixed duration of task i
α_f	Set-up time of family f
β_i	The variable duration of task i
ρ_{is}	Mass balance coefficient for consumption of state s in task i
$\bar{\rho}_{is}$	Mass balance coefficient for production of state s from task i
SO_s	Initial amount of state s
TC_c	Maximum timing of maintenance event c
B_i^{min}	Lower bounds on batch size of task i
B_i^{max}	Upper bounds on batch size of task i
θ_c	Maximum backward timing of maintenance event c
$cl_{ii'}$	The changeover time required between task i and i'
$sl_{ff'}$	The changeover time required between production family f and f'
Γ	Maximum amount of time allowed between cleaning events c
γ_f	Changeover time required between tasks within product family f
C_s^{min}	The minimum inventory level of state s
C_s^{max}	The maximum inventory level of state s
SC_s	Inventory storage costs of state s for one planning interval
HC_s	Material handling costs for state s
TC	The relative cost of 1 hour of changeover time
BC_s	Backlog penalty cost for product state s
λ_s	Cost of purchasing material state s
η_s	Revenue from the sale of one weight equivalent of product state s
$Prod_{st}$	The specified minimum product target for product state s
D_{st}	Demand for product state s in planning interval t

Binary Variables

Ws_{in}	=1 if task i starts processing at event point n
Wf_{in}	=1 if task i finishes processing at/or before event point n
Y_{cn}	=1 if maintenance event c occurs at event point n
Vs_{fn}	=1 if product family f starts processing at event point n
Vf_{fn}	=1 if product family f finished processing at/or before event point n
YYP_{ijgt}	=1 if task i is active on unit j in grouping g in planning interval t
YP_{fjgt}	=1 if product family f is active in planning interval t
PG_{gt}	=1 if production group g is active in planning interval t
$ZP_{ff'jgt}$	=1 if family f' is sequenced after family f in planning interval t
$ZZP_{ff'jgt}$	=1 if the link between product family f and f' in planning interval t

Continuous Variables

Z	Production throughput of production schedule
T_n	Time that corresponds to event point n
Ts_{in}	Start time of task i at event point n
Tf_{in}	Finish time of task i at event point n
Tp_{in}	Duration of task i that starts at event point n
Ts_{fn}	Start time of product family f at event point n
Tf_{fn}	Finish time of product family f at event point n
Tp_{fn}	Duration of product family f that starts at event point n
Bs_{in}	Batch size of task i that starts at event point n
Bp_{in}	Batch size of task i that is processing at event point n
Bf_{in}	Batch size of task i that ends at or before event point n
B^l_{isn}	Amount of state s used as an input for task i at event point n
B^O_{isn}	Amount of state s produced as an output from task i at event point n
S_{sn}	Amount of state s available at event point n
SS_{sn}	Amount of state s sold to market at event point n
T_{cn}	Timing assignment variable for cleaning event c at event point n
$\bar{T}_{cn}$	Equation slack variable for cleaning event c at event point n
$\hat{T}_{cn}$	Optimized backward timing of maintenance event c at event point n
Tfc	Completion time of the final cleaning event c
E_n	=1 if any cleaning event c occurs at event point n

Ga_{gn}	=1 if production group g is active at event point n
Fa_{fn}	=1 of product family f is active at event point n
Ff_{fn}	=1 if product family f finishes at event point n
$CO_{ff'n}$	=1 if product f precedes family f' on at event point n
$CO_{ii'n}$	=1 if task i precedes task i' on at event point n
FP_{ijgt}	The batch size of task i on unit j in group g at interval t
PT_{ijgt}	The processing of task i on unit j in group g at interval t
$Sale_{st}$	Amount of product state s sold to market in planning interval t
MO_{st}	Purchasing amount of state s in planning interval t
$Sale_{st}$	Amount of product state s sold to market in planning interval t
PGT_{jgt}	The processing time of production group g on unit j in interval t
$ICTV_{fjgt}$	The amount of transition time between tasks $i \in I_f$ in planning interval t
$TRNP_{jgt}$	The amount of changeover time between sequenced product families
HH_t	The balance of available processing time in planning interval t
BL_{st}	Amount of backlog of state s in planning interval t

List of References

Castro, P. M., Dogan, M. E., and Grossmann, I. E. (2008). Simultaneous batching and scheduling of single stage batch plants with parallel units. *AIChE Journal*, **54**(1), 183–193.

Chen, J.-S. (2008). Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan. *European Journal of Operational Research*, **1290**, 90–102.

Davies, K. M. (2008). Milp formulations for optimal steady-state buffer levels and flexible maintenance scheduling. Master's Thesis, McMaster, 280 Main Street West, Hamilton, Ontario L8S 4L8.

Dedopoulos, I. T. and Shah, N. (1995). Optimal short-term scheduling of maintenance and production for multipurpose plants. *Industrial & Engineering Chemistry Research*, **34**, 192–201.

Erdirik-Dogan, M. and Grossmann, I. E. (2007). Planning models for parallel batch reactors with sequence-dependent changeovers. *AIChE Journal*, **53**(9), 2284–2300.

Floudas, C. A. and Lin, X. (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers and Chemical Engineering*, **28**(11), 2109 – 2129.

Ierapetritou, M. G. and Floudas, C. A. (1998). Effective continuous-time formulation for short-term scheduling. 1. multipurpose batch processes. *Industrial & Engineering Chemistry Research*, **37**(11), 4341 – 4359.

Janak, S. L., Floudas, C. A., and Vormbrock, N. (2006). Production scheduling of a large-scale industrial batch plant. 1. short-term and medium-term scheduling. *Industrial & Engineering Chemistry Research*, **25**, 8234.

Janak, S. L., Lin, X., and 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. *Industrial & Engineering Chemistry Research*, **43**, 2516–2533.

Kondili, E., Pantelides, C. C., and Sargent, R. W. (1993). A general algorithm for short-term scheduling of batch operations-i. milp formulations. *Computers and Chemical Engineering*, **17**(2), 211 – 227.

Lee, C.-Y. (1996). Machine scheduling with an availability constraint. *Journal of Global Optimization*, **9**, 395–416.

Liao, C. J. and Chen, W. J. (2003). Single-machine scheduling with periodic maintenance and nonresumable jobs. *Computers & Operations Research*, **30**, 1335–1347.

Lin, X., Floudas, C. A., Modi, S., and Juhasz, N. M. (2002). Continuous-time optimization approach for medium-range production scheduling of a multiproduct batch plant. *Industrial & Engineering Chemistry Research*, **41**, 3884–3906.

Liu, S., Pinto, J. M., and Papageorgiou, L. G. (2008). A tsp-based milp model for medium-term planning of single-stage continuous multiproduct plants. *Industrial & Engineering Chemistry Research*, **47**, 7733–7743.

Maravelias, C. T. (2005). Mixed-time representation for state-task network models. *Industrial & Engineering Chemistry Research*, **44**, 9129–9245.

Maravelias, C. T. and Grossmann, I. E. (2003a). *A General Continuous State Task Network Formulation for Short Term Scheduling of Multipurpose Batch Plants with Due Dates*, Vol. 15A. Elsevier.

Maravelias, C. T. and Grossmann, I. E. (2003b). A new general continuous-time state task network formulation for short-term scheduling of multipurpose batch plants. *Industrial & Engineering Chemistry Research*, **42**(13), 3056–3074.

Maravelias, C. T. and Papalamprou, K. (2009). Polyhedral results for discrete-time production planning mip formulations for continuous processes. *Computers and Chemical Engineering*, **33**(11), 1890 – 1904.

McDonald, C. M. and Karimi, I. A. (1997). Planning and scheduling of parallel semi-continuous processes. 1. production planning. *Industrial & Engineering Chemistry Research*, **36**, 2691 – 2700.

Méndez, C. A., Cerdà, J., Grossmann, I. E., Harjunkoski, I., and Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers and Chemical Engineering*, **30**, 913 – 946.

Mockus, L. and Reklaitis, G. V. (1999). Continuous time representation approach to batch and continuous process scheduling. 1. minlp formulation. *Industrial & Engineering Chemistry Research*, **38**, 197–203.

Qi, X., Chen, T., and Tu, F. (1999). Scheduling the maintenance on a single machine. *Journal of Operational Research Society*, **50**, 1071–1078.

Shah, N., Pantelides, C. C., and Sargent, R. W. (1993). A general algorithm for short-term scheduling of batch operations-ii. computational issues. *Computers and Chemical Engineering*, **12**(2), 229 – 244.

Shaik, M. A., Janak, S. L., and Floudas, C. A. (2006). Continuous-time models for short-term scheduling of multipurpose batch plants: A comparative study. *Industrial & Engineering Chemistry Research*, **45**, 6190–6209.

Sherali, H. D. and Smith, J. C. (2001). Improving discrete model representations via symmetry considerations. *Management Science*, **47**(10), 1396–1407.

Sung, C. and Maravelias, C. T. (2007). An attainable region approach for production planning of multiproduct processes. *AIChE Journal*, **53**(5), 1298–1315.

Sung, C. and Maravelias, C. T. (2008). A mixed-integer programming formulation for the general capacitated lot-sizing problem. *Computers and Chemical Engineering*, **32**, 244–259.

Sung, C. and Maravelias, C. T. (2009). A projection-based method for production planning of multiproduct facilities. *AIChE Journal*, **55**(10), 2614–2630.

Verderame, P. M. and Floudas, C. A. (2008). Integrated operation planning and medium-term scheduling for large-scale industrial batch plants. *Industrial & Engineering Chemistry Research*, **47**, 4845–4860.

Verderame, P. M. and Floudas, C. A. (2009). Operational planning framework for multisite production and distribution networks. *Computers & Chemical Engineering*, **33**(5), 1036–1050.

Appendix A

Assignment Constraint Derivation

This derivation follows that given in [Maravelias and Grossmann, 2003b]. The logical expressions (A - C) and integer expression (D) can be used to derive the core assignment constraint of the continuous time state-task-network. Logical expressions (A) and (B) can be converted into integer equations (A*) and (B*), respectively:

$$Zs_{j,n} = \sum_{i \in I_j} Ws_{in} \qquad \forall j,\ n \tag{A*}$$

$$Zf_{j,n} = \sum_{i \in I_j} Wf_{in} \qquad \forall j,\ n \tag{B*}$$

Through the following series of equivalent representations and variable replacements, core assignment condition (C) is converted to constraint (3.21):

$$(Zs_{j,n} \Rightarrow \neg Zp_{j,n}) \Leftrightarrow$$

$$(\neg Zs_{j,n} \vee \neg Zp_{j,n}) \Leftrightarrow$$

$$((1 - Zs_{j,n}) + (1 - Zp_{j,n}) \geq 1) \Leftrightarrow$$

$$\left((1 - Zs_{j,n}) + \left(1 - \left(\sum_{n' < n} Zs_{j,n'} - \sum_{n' \leq n} Zf_{j,n'} \right) \right) \geq 1 \right) \Leftrightarrow$$

$$
\begin{aligned}
\left(2 - \sum_{n' \le n} Zs_{j,n'} - \sum_{n' \le n} Zf_{j,n'} \ge 1\right) &\Leftrightarrow \\
\left(\sum_{n' \le n} \left(Zs_{j,n'} - Zf_{j,n'}\right) \le 1\right) &\Leftrightarrow \\
\sum_{n' \le n} \left(\sum_{i \in I_j} Ws_{in'} - \sum_{i \in I_j} Wf_{in'}\right) \le 1 &\Leftrightarrow \\
\sum_{n' \le n} \sum_{i \in I_j} \left(Ws_{in'} - Wf_{in'}\right) \le 1 \quad \forall j,\ n &
\end{aligned}
$$

Appendix B

Material Delivery Methodology

The incorporation of product order due dates and material receipts was originally proposed by Maravelias and Grossmann [2003a], where orders or deliveries represent stationary events. Stationary events are described as those with fixed timing that may not be altered. There exists a set K composed of events k of product orders that must be met, or deliveries to be received. Let the fixed timing of event $k \in K$ be given as parameter TK_k and associate with each event the amount due (delivered) as parameter AD_k. Let set K_s and L_s represent the set of orders and deliveries that correspond to state s. Binary variable Y_{kn} is defined as 1 if event k is assigned to event point n and zero otherwise.

$$T_{kn} = TK_k Y_{kn} \qquad \forall k,\ n \tag{B.0.1}$$

The assignment of event k to any given event point n is given by constraint (B.0.1); which states if event k occurs at event point n, $Y_{kn} = 1$, then variable T_{kn} will assume the value of parameter TK_k. T_{kn} is defined as the timing of event k at event point n.

$$T_n = T_{kn} + \bar{T}_{kn} \qquad \forall k,\ n \tag{B.0.2}$$

$$\bar{T}_{kn} \leq H\left(1 - Y_{kn}\right) \qquad \forall k,\ n \tag{B.0.3}$$

When event k is assigned to event point n constraint (B.0.2) forces the timing of event point n to equal the timing of event k. $\bar{T}_{kn}$ is a slack variable introduced to relax constraint (B.0.2) when event k is not assigned to event point n. When event k occurs at event point n the slack variable must equate to zero to enforce the exact timing of constraint (B.0.2). This requirement is enforced through constraint (B.0.3).

$$\sum_{n} Y_{kn} = 1 \qquad \forall k \tag{B.0.4}$$

Every event k must occur and coincide with an event point n, as given by constraint (B.0.4). It is noted multiple events can occur at the same event point.

$$S_{s,n} + SS_{s.n} = S_{s,n-1} + \sum_{k \in L_s} AD_{kn} - \sum_{k \in K_s} AD_{kn} + \sum_{i \in O_s} B^{O}_{i,s,n} - \sum_{i \in I_s} B^{I}_{i,s,n} \quad \forall s,\ n \tag{B.0.5}$$

The original material balance constraint given as constraint (3.42) is modified to allow for delivery and receipt of materials and is now represented as constraint (B.0.5).

$$AD_{k.n} = AD_k Y_{kn} \qquad \forall k,\ n \tag{B.0.6}$$

Variable AD_{kn} is the amount of order (delivery) event k at event point n and is enforced through constraint (B.0.6).

Appendix C

Case Study Data

C.1 Model comparison data

Product state demands and associated processing data for model comparison case study in Section 4.1. β given in hours per weight equivalents of task batch size, demand given in weight equivalents.

Table C.1: Production data for model comparison case study.

State	Task	Family	Unit	β	Demand
Ps17	P17	F8	U2	0.3936	13.34
Ps18	P18	F4	U1	0.4261	3.52
Ps20	P20	F4	U2	0.4437	2.82
Ps19	P19	F4	U3	0.4502	2.78
Ps31	P31	F16	U1	0.3770	7.29
Ps33	P33	F16	U2	0.4006	4.37
Ps32	P32	F16	U3	0.3749	7.33
Ps39	P39	F23	U4	0.5249	28.58

C.2 Industrial Scheduling Case Study 1

Inventory and product state demands for industrial scheduling case study 1 in Section 4.2. β given in hours per weight equivalents of task batch size, demand given in weight equivalents.

Table C.2: Planned production data for industrial scheduling case study 1

State	Task	Family	Unit	β	Demand
Ps17	P17	F8	U2	0.3936	13.34
Ps18	P18	F9	U1	0.4261	3.52
Ps20	P20	F9	U2	0.4437	2.82
Ps19	P19	F9	U3	0.4502	2.78
Ps31	P31	F16	U1	0.3770	7.29
Ps33	P33	F16	U2	0.4006	4.37
Ps32	P32	F16	U3	0.3749	7.33
Ps39	P39	F23	U4	0.5249	28.58
Ps26	P26	F14	U1	0.3945	6.34
Ps28	P28	F14	U2	0.3370	2.22
Ps37	P37	F28	U4	0.5249	19.05
Ps2	P2	F1	U3	0.3976	5.66
Ps3	P3	F1	U2	0.3974	5.66
Ps1	P1	F1	U1	0.3846	7.15
Ps4	P4	F2	U1	0.4118	6.68
Ps29	P29	F15	U1	0.4079	8.58
Ps30	P30	F15	U2	0.4022	4.97
Ps38	P38	F26	U4	0.5249	38.10
Ps47	P47	F20	U1	0.3904	10.89
Ps48	P48	F20	U2	0.3851	6.49
Ps49	P49	F20	U3	0.3976	5.03

Table C.3: Initial inventory of material states for industrial scheduling case study 1, data given as equivalent weight units.

State	Inventory	State	Inventory
Rm1	0.127	Rm29	1.65
Rm2	0.064	Rm30	0.32
Rm3	0.007	Rm31	0.43
Rm4	0.095	Rm32	0.04
Rm5	22.14	Rm33	0.12
Rm6	0.62	Rm34	2.42
Rm7	2.85	Rm35	0.00
Rm8	7.12	Rm36	5.55
Rm9	50.00	Rm37	4.866
Rm10	14.36	Rm38	2.14
Rm11	3.97	Rm39	1.41
Rm12	14.51	Rm40	34.17
Rm13	25.03	Rm41	7.73
Rm14	1.35	Rm42	0.66
Rm15	0	Rm43	4.34
Rm16	0.525	Rm44	1.65
Rm17	13.12	Rm45	10.41
Rm18	0.04	Rm46	18.03
Rm19	0.20	Rm47	0.58
Rm20	0.025	Rm48	1.59
Rm21	0.75	Rm49	7.33
Rm22	3.30	Rm50	2.65
Rm23	1.83	Rm51	0.43
Rm24	0.99	Rm52	8.38
Rm25	2.69	Rm53	2.48
Rm26	0.01	Rm54	0.41
Rm27	21.01	Rm55	13.09
Rm28	41.5	Rm56	1.36

C.3 Industrial Scheduling Case Study 2

Inventory and product state demands for industrial scheduling case study 2 in Section 4.3. β given in hours per weight equivalents of task batch size, demand given in weight equivalents.

Table C.4: Planned production data for case study 2

State	Task	Family	Unit	β	Demand
Ps8	P8	F6	U1	3.6235	4.53
Ps9	P9	F6	U2	2.8775	15.11
Ps26	P26	F14	U1	2.8164	6.34
Ps28	P28	F14	U2	2.9670	3.71
Ps37	P37	F28	U4	1.9848	23.82
Ps34	P34	F23	U4	1.8054	28.58
Ps1	P1	F1	U1	2.8599	7.15
Ps3	P3	F1	U2	2.5163	5.66
Ps2	P2	F1	U3	2.4638	8.62
Ps29	P29	F15	U1	2.4513	8.58
Ps30	P30	F15	U2	2.7365	2.74
Ps38	P38	F26	U4	1.9487	42.87
Ps31	P31	F16	U1	2.9615	7.40
Ps33	P33	F16	U2	2.6885	8.74
Ps32	P32	F16	U3	2.5675	10.27
Ps23	P23	F11	U2	2.6471	7.28

Table C.5: Initial inventory of material states for industrial scheduling case study 2, data in in equivalent weight units.

State	Inventory	State	Inventory
Rm1	0.13	Rm29	0.72
Rm2	0.01	Rm30	0.27
Rm3	0.02	Rm31	0.58
Rm4	0.06	Rm32	0.06
Rm5	21.68	Rm33	0.11
Rm6	3.29	Rm34	3.42
Rm7	2.82	Rm35	0.00
Rm8	7.74	Rm36	4.1
Rm9	46.99	Rm37	2.84
Rm10	13.73	Rm38	0.839
Rm11	4.3	Rm39	2.87
Rm12	17.23	Rm40	19.78
Rm13	10.48	Rm41	7.69
Rm14	1.33	Rm42	0.89
Rm15	0	Rm43	3.66
Rm16	0.87	Rm44	2.54
Rm17	7.53	Rm45	24.56
Rm18	0.05	Rm46	8.44
Rm19	0.08	Rm47	0.18
Rm20	0.08	Rm48	0.56
Rm21	1.76	Rm49	6.98
Rm22	1.98	Rm50	2.57
Rm23	2.18	Rm51	0.43
Rm24	0.47	Rm52	1.87
Rm25	2.15	Rm53	3.37
Rm26	0.05	Rm54	0.29
Rm27	12.14	Rm55	6.93
Rm28	59.18	Rm56	1.7

C.4 Industrial Scheduling Case Study 3

Inventory and product state demands for industrial scheduling case study 3 in Section 4.4. β given in hours per weight equivalents of task batch size, demand given in weight equivalents.

Table C.6: Production targets for industrial scheduling case study 3

State	Task	Family	Unit	β	Demand
Ps8	P8	F6	U1	3.6235	4.53
Ps9	P9	F6	U2	2.8775	15.11
Ps26	P26	F14	U1	2.8164	6.34
Ps28	P28	F14	U2	2.9670	3.71
Ps37	P37	F28	U4	1.9848	23.82
Ps34	P34	F23	U4	1.8054	28.58
Ps10	P10	F4	U1	2.5034	5.63
Ps11	P11	F4	U3	2.4474	14.07
Ps12	P12	F4	U2	1.4426	3.25
Ps1	P1	F1	U1	2.8599	7.15
Ps3	P3	F1	U2	2.5163	5.66
Ps2	P2	F1	U3	2.4638	8.62
Ps29	P29	F15	U1	2.4513	8.58
Ps30	P30	F15	U2	2.7365	2.74
Ps38	P38	F26	U4	1.9487	42.87

Table C.7: Initial inventory of material states for industrial scheduling case study 3, data in in equivalent weight units.

State	Inventory	State	Inventory
Rm1	0.13	Rm29	0.72
Rm2	0.01	Rm30	0.27
Rm3	0.02	Rm31	0.58
Rm4	0.06	Rm32	0.06
Rm5	21.68	Rm33	0.11
Rm6	3.29	Rm34	3.42
Rm7	2.82	Rm35	5.91
Rm8	7.74	Rm36	4.1
Rm9	46.99	Rm37	2.84
Rm10	13.73	Rm38	0.839
Rm11	4.3	Rm39	2.87
Rm12	17.23	Rm40	19.78
Rm13	10.48	Rm41	7.69
Rm14	1.33	Rm42	0.89
Rm15	0	Rm43	3.66
Rm16	0.87	Rm44	2.54
Rm17	7.53	Rm45	24.56
Rm18	0.05	Rm46	0.00
Rm19	0.08	Rm47	0.18
Rm20	0.08	Rm48	0.56
Rm21	1.76	Rm49	6.98
Rm22	1.98	Rm50	2.57
Rm23	2.18	Rm51	0.43
Rm24	0.47	Rm52	1.87
Rm25	2.15	Rm53	3.37
Rm26	0.05	Rm54	0.29
Rm27	12.14	Rm55	6.93
Rm28	59.18	Rm56	1.7

C.5 Industrial Scheduling Case Study 4

Inventory and product state demands for industrial scheduling case study 4 given in section 4.5. β given in hours per weight equivalents of task batch size, demand given in weight equivalents.

Table C.8: Planned production data for industrial scheduling case study 4

State	Task	Family	Unit	β	Demand
Ps43	P43	F18	U2	0.417	11.982
Ps44	P44	F18	U3	0.361	7.62
Ps21	P21	F10	U1	0.484	7.228
Ps5	P5	F3	U1	0.405	7.403
Ps6	P6	F3	U3	0.351	12.125
Ps10	P10	F4	U1	0.382	9.152
Ps11	P11	F4	U3	0.396	11.256
Ps19	P19	F9	U3	0.396	6.94
Ps7	P7	F3	U2	0.363	4.817
Ps23	P23	F11	U2	0.378	7.28
Ps42	P42	F18	U1	0.367	7.484
Ps17	P17	F8	U2	0.375	13.336
Ps45	P45	F19	U1	0.523	5.254
Ps46	P46	F19	U2	0.348	7.185
Ps35	P35	F24	U4	0.525	38.102
Ps34	P34	F25	U4	0.519	42.864
Ps36	P36	F27	U4	0.551	38.102
Ps49	P49	F20	U3	0.398	5.029

Table C.9: Initial inventory of material states for industrial scheduling case study 4, data in in equivalent weight units.

State	Inventory	State	Inventory
Rm1	0	Rm29	0.90
Rm2	0.02	Rm30	0.27
Rm3	0.03	Rm31	0.63
Rm4	0.07	Rm32	0.06
Rm5	26.83	Rm33	0.12
Rm6	2.87	Rm34	3.88
Rm7	3.50	Rm35	13.90
Rm8	12.69	Rm36	4.71
Rm9	712.00	Rm37	4.00
Rm10	17.23	Rm38	1.10
Rm11	3.428	Rm39	3.44
Rm12	14.51	Rm40	21.66
Rm13	13.80	Rm41	12.22
Rm14	1.98	Rm42	1.12
Rm15	0.28	Rm43	1.85
Rm16	0.95	Rm44	2.66
Rm17	13.20	Rm45	3.18
Rm18	0.05	Rm46	30.83
Rm19	0.13	Rm47	0.19
Rm20	0.13	Rm48	71.03
Rm21	2.37	Rm49	8.41
Rm22	1.99	Rm50	1.57
Rm23	2.18	Rm51	0.57
Rm24	0.48	Rm52	1.73
Rm25	2.82	Rm53	3.58
Rm26	0.06	Rm54	0.30
Rm27	18.48	Rm55	6.93
Rm28	54.59	Rm56	2.15

C.6 Policy Evaluation Case Study

Inventory and product state demands for computer aided policy evaluation scheduling case study in Section 4.6. β given in hours per weight equivalents of task batch size, demand given in weight equivalents. Data is given for both operational policy S1 and operational policy S2.

Table C.10: Planned production data for policy evaluation case study - operational strategy S1. β given in hours per weight equivalents of task batch size, demand given in weight equivalents

State	Task	Family	Unit	β	Demand
Ps17	P17	F8	U2	0.3936	13.34
Ps18	P18	F9	U1	0.4261	3.52
Ps20	P20	F9	U2	0.4437	2.82
Ps19	P19	F9	U3	0.4502	2.78
Ps31	P31	F16	U1	0.6283	7.29
Ps33	P33	F16	U2	0.6676	4.37
Ps32	P32	F16	U3	0.6248	7.33
Ps39	P39	F23	U4	0.5249	28.58
Ps26	P26	F14	U1	0.6575	6.34
Ps28	P28	F14	U2	0.5616	2.22
Ps37	P37	F28	U4	0.5249	19.05
Ps2	P2	F1	U3	0.3976	5.66
Ps3	P3	F1	U2	0.3974	5.66
Ps1	P1	F1	U1	0.3846	7.15
Ps4	P4	F2	U1	0.4118	6.68
Ps29	P29	F15	U1	0.6798	8.58
Ps30	P30	F15	U2	0.6703	4.97
Ps38	P38	F26	U4	0.5249	38.10
Ps47	P47	F20	U1	0.3904	10.89
Ps48	P48	F20	U2	0.3851	6.49
Ps49	P49	F20	U3	0.3976	5.03

Table C.11: Planned production data for policy evaluation case study - operational strategy S2. β given in hours per weight equivalents of task batch size, demand given in weight equivalents

State	Task	Family	Unit	β	Demand
Ps17	P17	F8	U2	0.6561	13.34
Ps18	P18	F9	U1	0.7102	3.52
Ps20	P20	F9	U2	0.7396	2.82
Ps19	P19	F9	U3	0.7505	2.78
Ps31	P31	F16	U1	0.6284	7.29
Ps33	P33	F16	U2	0.6677	4.37
Ps32	P32	F16	U3	0.6249	7.33
Ps39	P39	F23	U4	0.5249	28.58
Ps26	P26	F14	U1	0.6576	6.34
Ps28	P28	F14	U2	0.5618	2.22
Ps37	P37	F28	U4	0.5249	19.05
Ps2	P2	F1	U3	0.6627	5.66
Ps3	P3	F1	U2	0.6624	5.66
Ps1	P1	F1	U1	0.6411	7.15
Ps4	P4	F2	U1	0.6864	6.68
Ps29	P29	F15	U1	0.6800	8.58
Ps30	P30	F15	U2	0.6704	4.97
Ps38	P38	F26	U4	0.5249	38.10
Ps47	P47	F20	U1	0.6506	10.89
Ps48	P48	F20	U2	0.6419	6.49
Ps49	P49	F20	U3	0.6627	5.03

C.7 Production Planning Data

General data for production planning case studies and state-task network information, including: production group assignment (table C.12), sequence dependent changeover chart (table C.13), product state demands for the 12 week horizon (table C.14), product family-material formulation (table C.15) and general STN formulation data (tables C.16 and C.17).

Table C.12: Production grouping assignment for product families

	P1	P2	P3	P4	P5	P6
F1	no	no	yes	no	no	no
F2	no	no	yes	no	no	no
F15	no	no	yes	no	no	no
F3	no	yes	no	no	no	no
F22	no	yes	no	no	no	no
F4	yes	no	no	no	no	no
F5	yes	no	no	no	no	no
F17	yes	no	no	no	no	no
F7	no	no	no	yes	no	no
F12	no	no	no	yes	no	no
F13	no	no	no	yes	no	no
F14	no	no	no	yes	no	no
F16	yes	yes	yes	yes	yes	yes
F6	yes	yes	yes	yes	yes	yes
F8	yes	yes	yes	no	yes	yes
F9	no	no	yes	no	yes	no
F10	no	no	yes	no	yes	no
F11	yes	yes	yes	no	yes	yes
F18	yes	yes	yes	yes	yes	yes
F19	yes	yes	yes	no	yes	yes
F20	yes	yes	yes	yes	yes	yes
F21	yes	yes	yes	yes	yes	yes
F24	yes	no	no	no	no	no
F25	no	yes	no	no	no	no
F26	no	no	yes	no	no	no
F27	no	no	no	no	yes	no
F23	no	no	no	no	no	yes
F28	no	no	no	yes	no	no

Table C.13: Sequence dependent changeover duration, given in hours, between product families

F	1	2	3	6	4	7	8	9	11	12	13	14	15	16	17	18	19	20	22	5	10	21	28	23	25	24	26	27
1	$\frac{1}{4}$	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1
2	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	$\frac{1}{4}$	1	1	1
6	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1
7	1	1	1	1	1	$\frac{1}{4}$	1	1	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	3	3	3	3	3	3	3	3	$\frac{1}{4}$	3	3	3	3	3	1	3	3	3	3	3	3	3	3	3	3	3	3	3
12	1	1	1	1	1	$\frac{1}{4}$	1	1	1	$\frac{1}{4}$	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	3	3	3	3	3	3	3	3	3	3	$\frac{1}{4}$	$\frac{1}{4}$	3	3	3	3	3	3	3	3	3	3	$\frac{1}{4}$	3	3	3	3	3
14	3	3	3	3	3	3	3	3	3	3	$\frac{1}{4}$	$\frac{1}{4}$	3	3	3	3	3	3	3	3	3	3	1	3	3	3	3	3
15	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1
17	3	3	3	3	3	3	3	3	1	3	3	3	3	3	$\frac{1}{4}$	3	3	3	3	3	3	3	3	3	3	3	3	3
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1
19	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	$\frac{1}{4}$	3	3	3	3	3	3	3	3	3	3	3
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1
22	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	$\frac{1}{4}$	1	1	$\frac{1}{4}$	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	$\frac{1}{4}$
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1
28	3	3	3	3	3	3	3	3	3	3	$\frac{1}{4}$	$\frac{1}{4}$	3	3	3	3	3	3	3	3	3	3	$\frac{1}{4}$	3	3	3	3	3
23	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1
25	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	$\frac{1}{4}$	1	1	$\frac{1}{4}$	1	1	1
24	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	$\frac{1}{4}$	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	$\frac{1}{4}$

Table C.14: Product demands over planning horizon intervals (weight equivelants)

Interval	1	2	3	4	5	6	7	8	9	10	11	12
Ps43	1.374	0.878	1.585	1.966	1.837	2.150	2.000	2.198	1.089	1.415	1.749	2.034
Ps44	0.479	0.180	0.659	1.033	1.272	0.988	1.302	0.299	0.718	1.123	0.988	1.078
Ps29	1.223	2.192	2.083	1.800	4.576	1.680	3.186	2.547	2.250	3.803	1.967	2.845
Ps30	1.702	0.612	1.121	2.253	2.290	2.792	2.370	2.676	0.968	2.272	1.696	2.541
Ps21	1.143	0.686	2.482	1.691	2.783	1.655	2.101	1.756	1.854	2.526	1.702	1.967
Ps4	0.987	0.152	0.864	1.230	2.308	1.531	1.644	2.101	1.912	1.727	1.408	2.203
Ps31	1.557	1.782	3.821	2.910	3.600	2.634	3.567	4.246	3.560	4.021	3.135	3.781
Ps5	1.735	2.685	4.801	3.190	4.311	3.629	4.888	5.011	3.792	5.777	4.485	4.888
Ps32	0.793	1.033	3.054	3.113	3.458	2.694	3.907	3.143	2.260	2.754	3.847	3.742
Ps6	2.799	0.464	1.527	4.999	6.032	4.356	5.688	5.134	3.024	4.521	4.116	4.116
Ps1	1.865	2.319	3.705	3.865	3.817	3.103	4.090	4.634	3.382	5.559	1.963	3.335
Ps14	1.063	1.633	1.981	2.264	2.722	1.829	2.878	3.567	1.992	3.447	2.322	2.010
Ps10	1.829	2.986	4.797	4.521	5.236	3.905	3.934	5.599	3.716	7.613	2.794	4.786
Ps26	0.991	2.061	2.010	3.175	2.776	1.579	2.787	3.807	2.297	4.213	2.391	2.780
Ps18	1.401	1.655	2.337	2.370	3.251	2.159	2.409	2.663	1.822	2.112	3.113	2.667
Ps2	2.904	1.497	4.206	4.461	6.856	3.413	6.257	4.371	4.895	3.158	4.610	4.730
Ps15	1.227	1.347	2.185	4.101	5.449	2.814	4.236	4.775	2.200	2.515	4.535	4.431
Ps11	1.527	2.440	3.293	3.413	6.002	4.371	5.778	4.191	4.251	4.101	5.838	3.787
Ps27	0.389	0.000	0.913	2.694	3.727	2.769	3.637	3.952	1.302	4.431	1.617	2.949
Ps19	1.129	0.095	1.524	1.470	0.912	2.300	3.973	1.592	0.871	1.470	1.211	2.490
Ps33	3.007	1.075	2.538	4.191	2.062	2.000	2.796	4.559	2.259	2.504	2.007	2.844
Ps7	1.374	1.347	3.892	4.048	3.416	2.388	4.028	3.402	2.477	2.483	4.232	2.694
Ps3	3.484	1.538	3.205	5.674	4.402	3.456	3.667	3.177	3.626	4.055	3.422	3.783
Ps16	1.422	1.524	1.497	3.116	2.347	1.449	1.354	2.089	2.177	1.456	2.157	1.830
Ps12	2.538	1.531	5.321	3.150	3.436	3.007	4.511	4.354	2.701	3.660	3.354	4.144
Ps28	0.456	0.748	1.286	3.293	1.510	1.483	1.470	2.000	1.089	1.558	1.857	1.613
Ps23	2.082	0.061	1.796	0.218	0.891	0.558	1.966	1.306	1.089	0.776	1.225	0.925
Ps20	1.690	0.667	1.041	2.596	1.182	0.667	1.874	1.855	1.874	0.857	1.947	1.819
Ps52	0.443	0.374	1.742	0.980	1.383	0.726	1.836	1.314	1.034	1.205	1.212	1.227
Ps13	0.871	0.889	1.843	1.430	2.007	0.686	1.575	1.742	1.063	1.172	1.528	2.409
Ps53	0.000	0.000	0.272	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Ps8	1.549	2.010	3.436	5.683	6.107	5.403	5.548	4.899	5.951	5.625	3.618	5.537
Ps9	1.823	1.871	2.422	4.599	5.579	3.252	4.742	5.729	3.171	3.552	3.742	4.736
Ps50	0.367	1.081	0.319	1.241	1.608	1.089	1.575	1.967	1.343	2.830	0.922	2.486
Ps51	0.095	0.694	0.585	1.252	1.150	1.184	1.830	0.980	0.885	0.953	1.647	1.463
Ps40	0.392	1.582	1.052	1.143	2.453	0.954	1.854	2.159	1.379	2.243	1.310	1.858
Ps41	1.442	0.361	0.578	2.068	2.619	1.878	1.932	2.960	1.068	0.728	1.817	1.531
Ps42	1.212	1.107	2.907	1.909	2.253	2.105	1.814	3.429	1.978	3.360	0.885	2.108
Ps17	2.708	1.674	2.830	2.742	3.905	2.667	4.463	3.422	2.014	1.708	2.715	2.790
Ps45	0.559	0.541	1.187	1.466	1.259	1.125	0.991	1.237	1.586	1.930	0.682	1.317
Ps46	0.095	0.265	0.299	1.041	0.435	0.864	0.633	0.959	0.626	0.612	0.476	0.966
Ps39	12.024	12.908	13.248	13.734	13.938	16.019	14.502	17.117	19.090	15.571	13.530	12.296
Ps35	15.222	18.662	19.197	19.518	19.129	22.055	20.451	24.261	25.661	22.541	19.148	17.933
Ps34	19.877	22.920	24.242	24.271	24.456	28.995	25.359	30.346	32.669	26.827	24.776	23.493
Ps38	16.145	19.284	19.975	19.839	20.402	23.435	21.092	25.010	26.876	22.871	19.372	18.332
Ps36	19.382	23.668	24.990	25.087	25.369	28.946	27.342	31.532	34.156	28.917	26.302	23.824
Ps37	10.682	12.063	12.480	12.383	12.898	15.309	13.666	16.893	18.653	15.533	12.519	12.500
Ps47	1.491	1.070	3.846	2.112	3.981	2.667	3.092	5.062	2.910	4.333	3.084	3.433
Ps48	1.150	1.789	0.660	1.136	1.279	2.470	2.660	2.313	0.946	2.300	2.164	1.864
Ps49	0.354	0.381	2.341	1.987	2.204	1.170	2.286	3.075	0.898	3.157	3.416	0.490

Table C.15: Product family production proportions for product families F1 to F14, given as % of batch size in weight equivalents

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
Rm5	0.13887	0.09558	0.12976	0.11276	0.08610	0.12783	0.08610	0.15105	0.14059	0.0576	0.12433	0.10176	0.10686	0.11337
Rm6	0.00439	0.00204	0.00472	0.00205	0.00157	0.00374	0.00157	0.00216	0.00489	0.00596	0.00370	0.00302	0.00320	0.00341
Rm7	0.00329	0.00680	0.00590	0.00264	0.00626	0.00488	0.00626	0.00432	0.00733	0.00535	0.00252	0.00357	0.00455	0.00477
Rm8			0.05898		0.06703	0.06099	0.06703		0.00087	0.07485	0.04267	0.03625		
Rm9	0.29467	0.30612	0.31259	0.29143	0.23481	0.33501	0.23481	0.30210	0.38421	0.25206	0.17977	0.26641	0.28653	0.31026
Rm10	0.01332	0.02245									0.16801			
Rm11								0.11329						
Rm12														
Rm13					0.17504		0.17504	0.14797		0.04812	0.03058	0.17166		
Rm14	0.00063	0.00051	0.00059	0.00059	0.01039	0.01220	0.01039		0.00070			0.00810	0.00051	0.00068
Rm17										0.05240	0.07913			
Rm19										0.00107				
Rm20			0.00088											
Rm21				0.00513										
Rm22						0.08538								
Rm25										0.01436			0.03101	0.03222
Rm26				0.00021									0.00843	0.00869
Rm27			0.05072	0.04101	0.08880		0.08880		0.06112	0.05561		0.07059	0.09001	0.09547
Rm28	0.27586	0.26531	0.20938	0.21235	0.17789	0.25370	0.17789	0.17725	0.17220	0.21387	0.20161	0.14007	0.27305	0.28128
Rm29			0.00177		0.00356		0.00356				0.00168	0.00330	0.00219	0.00239
Rm30						0.00553								
Rm31									0.00978					
Rm35	0.00031	0.00034		0.24603							0.05998			
Rm36	0.00266	0.00289	0.00310	0.00205	0.00256	0.00309	0.00256	0.00879	0.00210	0.00306	0.00118	0.00206	0.00236	0.00239
Rm37	0.00611	0.00901	0.00472	0.00308	0.00470	0.00439	0.00470	0.00339	0.00559	0.00122	0.00554	0.00426	0.00455	0.00477
Rm38	0.00611	0.00476		0.00425				0.00339		0.00565	0.00790		0.00624	0.00648
Rm39			0.00516		0.00470	0.00439	0.00470		0.00506			0.00426		
Rm40	0.08887	0.08997	0.08847	0.07381	0.07998	0.09156	0.07998	0.08092	0.09780	0.08295	0.09140	0.07814	0.09001	0.09853
Rm41									0.10217	0.10434				
Rm42	0.00110		0.00354	0.00059	0.00242	0.00244	0.00242		0.00314	0.00061		0.00206	0.00236	0.00119
Rm44										0.02093				
Rm45			0.11501											
Rm46	0.15987	0.13503												
Rm55		0.05068			0.05422		0.05422					0.10299	0.08259	0.02983
Rm56	0.00392	0.00850	0.00472	0.00205		0.00488		0.00539	0.00244			0.00151	0.00556	0.00426

Table C.16: General state-task network planning data, β given in hours per weight equivalents of task batch size, inventory limits given in weight equivalents

Task	State	Family	β_i	C_s^{min}	C_s^{max}
P43	Ps43	F18	0.399	3.514	17.571
P44	Ps44	F18	0.419	1.878	9.39
P29	Ps29	F15	0.396	5.82	29.101
P30	Ps30	F15	0.399	4.075	20.377
P21	Ps21	F10	0.396	4.096	20.48
P4	Ps4	F2	0.396	3.289	16.445
P31	Ps31	F16	0.396	7.352	36.758
P5	Ps5	F3	0.396	9.107	45.537
P32	Ps32	F16	0.419	5.873	29.366
P6	Ps6	F3	0.419	8.126	40.631
P1	Ps1	F1	0.396	8.071	40.355
P14	Ps14	F7	0.396	5.064	25.321
P10	Ps10	F3	0.396	9.615	48.075
P26	Ps26	F14	0.396	5.44	27.201
P18	Ps18	F9	0.396	5.376	26.879
P2	Ps2	F1	0.419	8.721	43.606
P15	Ps15	F7	0.419	6.102	30.51
P11	Ps11	F3	0.419	8.811	44.054
P27	Ps27	F14	0.419	4.163	20.816
P19	Ps19	F9	0.419	3.204	16.018
P33	Ps33	F16	0.399	5.727	28.634
P7	Ps7	F3	0.399	6.412	32.06
P3	Ps3	F1	0.399	7.597	37.983
P16	Ps16	F7	0.399	3.909	19.544
P12	Ps12	F3	0.399	7.351	36.756
P28	Ps28	F14	0.399	2.947	14.736
P23	Ps23	F11	0.399	2.428	12.139
P20	Ps20	F9	0.399	3.261	16.307

Table C.17: General state-task network planning data, β given in hours per weight equivalents of task batch size, inventory limits given in weight equivalents

Task	State	Family	β_i	C_s^{min}	C_s^{max}
P52	Ps52	F22	0.396	2.216	11.081
P13	Ps13	F4	0.396	2.973	14.865
P53	Ps53	F22	0.399	0.02	0.101
P8	Ps8	F5	0.396	10.307	51.537
P9	Ps9	F5	0.399	7.81	39.051
P50	Ps50	F21	0.396	3.594	17.968
P51	Ps51	F21	0.399	2.556	12.782
P40	Ps40	F17	0.396	3.437	17.183
P41	Ps41	F17	0.399	3.671	18.353
P42	Ps42	F18	0.396	4.34	21.7
P17	Ps17	F8	0.399	6.288	31.439
P45	Ps45	F19	0.396	2.664	13.318
P46	Ps46	F19	0.399	1.821	9.107
P39	Ps39	F23	0.508	33.145	99.434
P35	Ps35	F24	0.508	45.438	227.189
P34	Ps34	F25	0.508	58.41	175.231
P38	Ps38	F26	0.508	47.904	143.713
P36	Ps36	F27	0.508	60.688	182.064
P37	Ps37	F28	0.508	30.748	92.244
P47	Ps47	F20	0.396	7.184	35.918
P48	Ps48	F20	0.399	3.882	19.409
P49	Ps49	F20	0.419	3.167	15.837

Appendix D

Combined Maintenance and Production Scheduling

Processing units may be required to be taken offline for inspection, modification or equipment maintenance. Equipment maintenance is typically a predefined operation planned in advance of the production schedule and is carried out by employees, maintenance crews or outside contractors. There may exist various requirements to the exact timing of maintenance events; outside contractors may require a dedicated time period to carry out maintenance tasks, while facility employees may be more flexible. Several extensions are proposed to address maintenance activities with fixed, flexible and recurring timing requirements. Fixed timing is defined as maintenance events that must occur at a specified time, whereas flexible timing defines a scenario in which the maintenance task can occur within a specified window of time. Maintenance activities with recurring timing are dependent on the timing of previously completed maintenance tasks; this scenario is common in the food processing industry where units are taken offline for cleaning periodically. The following section will describe a framework that incorporates all three maintenance types into the continuous-time scheduling model, which is based on the scheduling formulation with due dates presented by Maravelias and Grossmann [2003a].

Maintenance Events with Fixed Timing

Let C represent the set of all maintenance events that are planned to occur within the scheduling horizon. A maintenance event corresponds to a point in time at which maintenance commences in one or more process units. The planned start time of maintenance events $c \in C$ is defined through parameter TC_c. Every maintenance event is associated with one or more tasks that represent the type of maintenance to be conducted and belongs to the set of all tasks $i \in I$. As such, maintenance tasks participate with the previously described assignment and timing constraints to ensure the correct occurrence of the task. The duration of the maintenance task is prescribed by variable Tp_{in} through parameter α_i as in constraint (3.25). Let I_c represent the set of tasks i that are carried out for maintenance event c. We note that maintenance event c can be associated with more than one maintenance task.

Maintenance Event Timing Assignment

As the timing of event points is unknown prior to optimization, the formulation must be free to assign maintenance event c to any given event point n. This is accomplished with the introduction of binary variable Y_{cn}, which is defined as 1 if maintenance event c begins at event point n. If maintenance event c occurs at event point n, then the timing of the current event point must equate to the timing of the maintenance event:

$$T_n = TC_c\, Y_{cn} + \bar{T}_{cn} \qquad \forall c,\ n \qquad \text{(D.0.1)}$$

$$0 \leq \bar{T}_{cn} \leq H\,(1 - Y_{cn}) \qquad \forall c,\ n \qquad \text{(D.0.2)}$$

Constraints (D.0.1) and (D.0.2) enforce the timing of maintenance event c by equating the event point timing (T_n) to the start time of maintenance event c if maintenance event c occurs at event point n. $\bar{T}_{cn}$ relaxes constraint (D.0.1) if maintenance event c does not occur at event point n. The following additional constraints are imposed:

$$\sum_n Y_{cn} = 1 \qquad \forall c \qquad \text{(D.0.3)}$$

$$Y_{cn} \leq Ws_{in} \qquad \forall c,\ i \in I_c,\ n \qquad \text{(D.0.4)}$$

$$Y_{cn} \geq Ws_{in} \qquad \forall c,\ i \in I_c,\ n \qquad \text{(D.0.5)}$$

Constraint (D.0.3) enforces all maintenance events c to occur within the scheduling horizon, while constraint (D.0.4) requires the associated maintenance tasks $i \in I_c$ to occur at maintenance event point n and constraint (D.0.5) ensures that maintenance tasks occur only at maintenance event points. The above set of constraints will enforce the occurrence of all planned maintenance events and the associated tasks to be carried out at the defined timing.

The combined process and maintenance scheduling model is composed of the continuous time state-task network scheduling model augmented with constraints (D.0.1) to (D.0.5). We note that for maintenance tasks, batch size and material balance constraints are redundant. However, utility constraints may be relevant for maintenance tasks that require utilities. We remark also that the task assignment and start, process and finish time constraints described in the previous section ensure that a maintenance task cannot occur concurrently with a processing task on the same unit.

Process Wide Maintenance Tasks

A process wide maintenance task is defined as taking all processing units offline for maintenance. This may be required in industry for product quality or health concerns. This can be accomplished in one of two ways. The first method specifies a task for each unit, such is consistent with the treatment of tasks in the global event framework. This can pose issues if the process wide maintenance event requires the use of utility resources. The second method specifies a single task i and the inclusion of task i into set I_j for all units j will ensure that if maintenance task i is scheduled to occur no other processing tasks can occur. This method aids in both reducing the number of binary variables present in the above formulation and simplifies the treatment of utility resources for such process wide tasks.

Maintenance Events with Flexible Timing

The inherent flexibility in multipurpose batch plants can be exploited by optimizing the timing of maintenance events. Flexibility in the timing of maintenance events c can be incorporated into the above framework through introduction of forward and backward slack variables into constraint (D.0.1). The timing of maintenance event c is allowed to vary either backward or forward from the planned timing, but this allowance is limited. Parameters η_c and θ_c represent the maximum bound on the forward and backward timing slack.

Modified Maintenance Event Timing Constraints

$\hat{T}_{cn}$ and $\widetilde{T}_{cn}$ are introduced as the backward and forward timing slack variables for maintenance c at event point n, giving the modified maintenance timing constraint

$$T_n = TC_c\, Y_{cn} + \bar{T}_{cn} - \hat{T}_{cn} + \widetilde{T}_{cn} \qquad \forall c,\ n \qquad \text{(D.0.6)}$$

with associated slack variable bounds

$$0 \leq \widetilde{T}_{cn} \leq \eta_c\, Y_{cn} \qquad \forall c,\ n \qquad \text{(D.0.7)}$$

$$0 \leq \hat{T}_{cn} \leq \theta_c\, Y_{cn} \qquad \forall c,\ n. \qquad \text{(D.0.8)}$$

$\hat{T}_{cn}$ and $\widetilde{T}_{cn}$ are zero if maintenance event c does not occur at event point n.

Recurring Maintenance Events

The methodology for recurring maintenance events presented in Chapter 3 is concerned with the scheduling of a process wide cleaning activity. As such the methodology presented is in respect to a one-to-one mapping of the cleaning tasks i with the binary indicator variable Y_{in}. A more general scenario is the one-to-many mapping of a maintenance event c with one or more maintenance tasks i through binary indicator Y_{cn}. The summarized general recurring maintenance scheduling methodology is given as:

$$T_n = TC_c\, Y_{cn} + \bar{T}_{cn} - \sum_{c' \leq c} \sum_{n' \leq n} \hat{T}_{c'n'} \qquad \forall c,\ n \qquad \text{(D.0.9)}$$

$$0 \leq \bar{T}_{cn} \leq H\,(1 - Y_{cn}) \qquad \forall c,\ n \qquad \text{(D.0.10)}$$

$$0 \leq \hat{T}_{cn} \leq \theta_c\, Y_{cn} \qquad \forall c,\ n. \qquad \text{(D.0.11)}$$

$$\sum_n Y_{cn} = 1 \qquad \forall c \qquad \text{(D.0.12)}$$

$$Y_{cn} \leq Ws_{in} \qquad \forall c,\ i \in I_c,\ n \qquad \text{(D.0.13)}$$

$$Y_{cn} \geq Ws_{in} \qquad \forall c,\ i \in I_c,\ n \qquad \text{(D.0.14)}$$

$$Tfc \leq Tf_{in} + H\,(1 - Y_{cn}) \qquad \forall i \in |I_c|,\ n \qquad \text{(D.0.15)}$$

$$Tfc \geq Tf_{in} - H\,(1 - Y_{cn}) \qquad \forall i \in |I_c|,\ n \qquad \text{(D.0.16)}$$

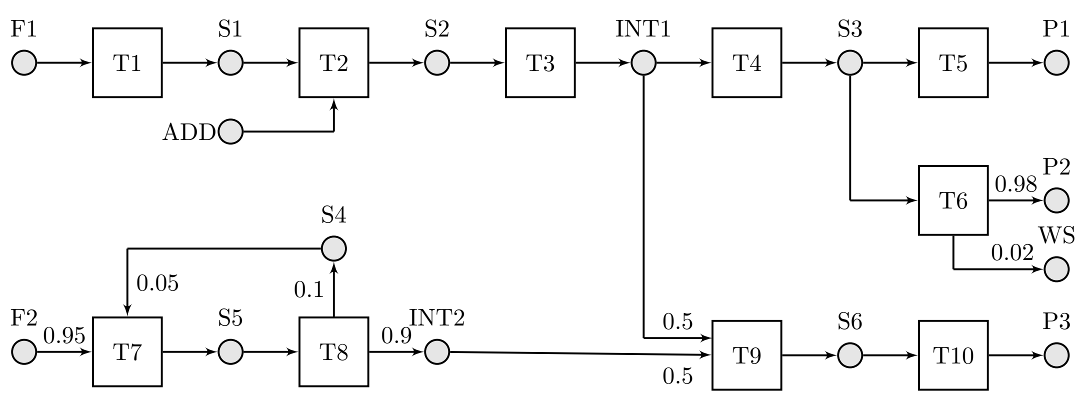
$$H - Tfc \leq \Gamma \qquad \text{(D.0.17)}$$

Composite Model Summary Four models have been defined in the above formulations and include: M - The general continuous-time global event scheduling model as presented by Maravelias and Grossmann [2003a]. M^{fix} - Maintenance scheduling with fixed timing, given as core model M with constraints (D.0.1) to (D.0.5). M^{flex} - Maintenance scheduling with flexible timing, given as core model M with constraints (D.0.2) to (D.0.8). M^{recur} - Recurring maintenance event scheduling, given as core model M with constraints (D.0.9) to (D.0.17).

D.1 Case Studies

In this section we use the proposed extensions to solve a modification to a problem taken from Maravelias and Grossmann [2003b]. The example is a multipurpose batch plant that produces multiple product states; the state-task network of the process is given in Figure D.1. The plant has 6 units (U1-U6) capable of performing 10 processing tasks (T1-T10); the unit-task assignment data is given in Table D.1. 14 material states are described; raw material states (F1-F2), intermediate states (S1-S6,INT1-INT2), product states (P1-P3) and a waste stream (WS). Multiple material handling policies are enforced; the data are given in Table D.2. Three utility streams are available for use; high pressure steam (HPS), low pressure steam (LPS) and cooling water (CW) with maximum availabilities given as 20, 40 and 25 kg/min, respectively.

Figure D.1: State-task network of example problem [Maravelias and Grossmann, 2003b]



Four case studies are presented to showcase the usage of the proposed framework. The first

Table D.1: Data for Case Studies (B^{max} in kg, α in hr, γ in kg/min and δ in kg/min per kg of batch)

Task	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	CU2	CU3
Unit	U1	U2	U3	U1	U4	U4	U5	U6	U5	U6	U2	U3
B^{max}	5	8	6	5	8	8	3	4	3	4	1	1
α	2	1	1	2	2	2	4	2	2	3	2	2
Utility	LPS	CW	LPS	HPS	LPS	HPS	CW	LPS	CW	CW	-	-
γ	3	4	4	3	8	4	5	5	5	3	-	-
δ	2	2	3	2	4	3	4	3	3	3	-	-

Table D.2: Storage limitations, initial state inventory, sales price and storage policy data for case study example problem

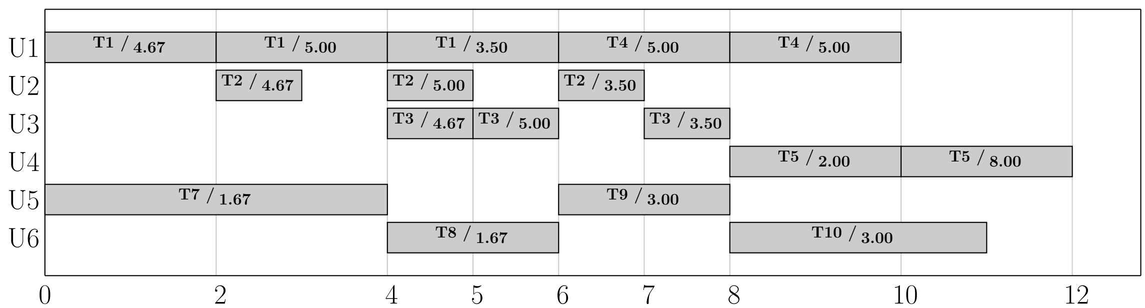
	F1	F2	S1	S2	S3	S4	S5	S6	INT1	INT2	P1	P2	P3
C_s(kg)	∞	∞	0	0	15	40	0	0	∞	∞	∞	∞	∞
SO_s(kg)	100	100	0	0	0	10	0	0	0	0	0	0	0
g_s(10^3 \$/kg)	-	-	-	-	-	-	-	-	-	-	1	1	1
Storage Policy	US	US	ZW	NIS	FS	FS	ZW	NIS	US	US	US	US	US

case study will examine the solution to the original problem as presented by Maravelias and Grossmann [2003b]. The second case study will include two maintenance events of fixed timing into the original problem. The third case study will allow flexibility to the timing of the maintenance presented in the second case study. The final case study considers the inclusion of a recurring process-wide maintenance event within the original problem with an extended time horizon. Finally, conclusions are drawn on the various results and computational efficiency observed with the above extensions. The model was coded in GAMS 23.6.2 and all case studies solved using CPLEX 12.2 with the optimality gap criteria set to 0%. The software was run on a 2.33 GHz Intel® Core™2 Quad Dell XPS desktop computer with 6 gigabytes of RAM running Windows Vista Ultimate 64-bit.

D.1.1 Case Study 1 - Original Problem

The original problem was solved with core model M with a scheduling horizon of 12 hours. The optimal solution needed 9 event points; the resulting model comprised 180 discrete variables, 1,574 continuous variables and 2,874 constraints. The relaxed LP solution of model M is given as \$19,470. The profitability of the production schedule was determined as \$13,000 and was obtained in 6.160 CPUs after exploring 1,543 nodes. 10 kg of product state P1 and 3kg of product state P3 are sold to market. A Gantt chart showing the optimal process schedule is given in Figure D.2, with processing tasks represented by rectangles (task identity and batch size listed within the rectangle). All material transfer policies are strictly adhered to, as is the maximum utility availability and limited intermediate storage restrictions. The solution presented is equivalent to that of Maravelias and Grossmann [2003b] and Janak *et al.* [2004].

Figure D.2: Optimal solution to case study 1.

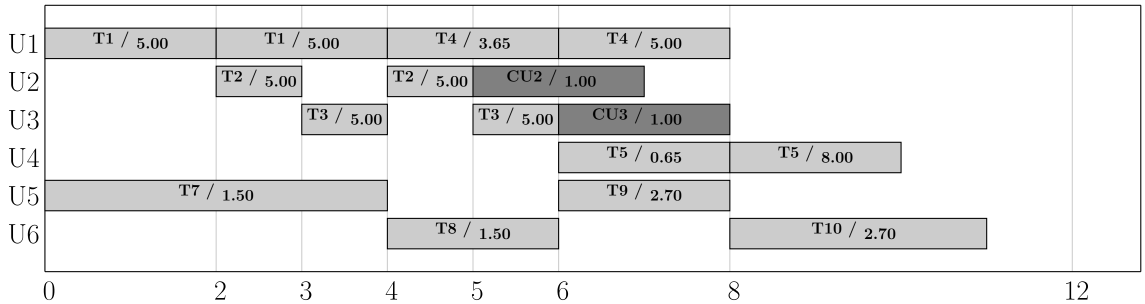


D.1.2 Case Study 2 - Fixed Maintenance Timing

Two maintenance events are planned to occur within the scheduling horizon. The first task ($CU2$) is a 2 hour maintenance performed on U2 ($\alpha_{CU2} = 2$) that is scheduled to begin at hour 5 ($TC_1 = 5$). The second task ($CU3$) is 2 hour clean ($\alpha_{CU3} = 2$) to begin on U3 at hour 6 ($TC_2 = 6$). Model M^{fix} was used to solve the combined scheduling and maintenance event optimization. The optimal solution is determined as \$11,350 and is obtained with 8 event points in 1.40 CPUs after exploring 105 nodes. At 8 event points, model M^{fix} has 208 binary variables, 1,636 continuous variables and 2,983 constraints; the optimal solution is depicted in Figure D.3. 8.65 kg of P1 and 2.7 kg of P3 are sold to market. Maintenance tasks $CU2$ and $CU3$ limit the available processing time of units U2 and U3. As units U2 an U3 process tasks related to the production of INT1, a reduction in the amount of product manufactured is exhibited.

To compare the computational performance of model M^{fix}, the base model M was re-

Figure D.3: Gantt chart of optimal solution for case study 2



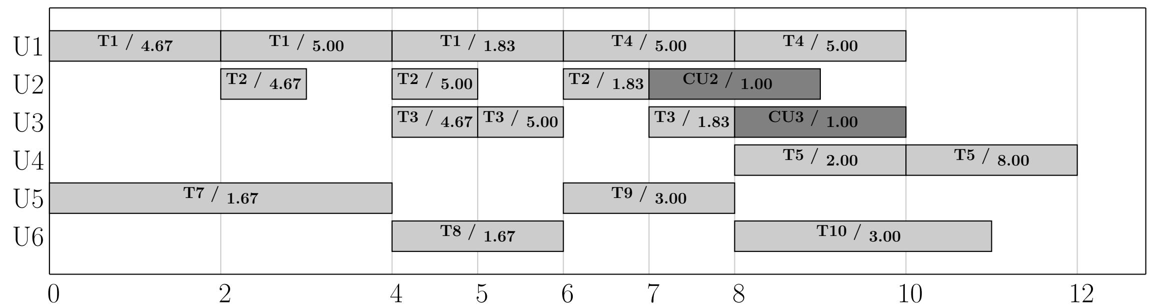
solved using only 8 event points. The resulting solution is suboptimal; we generate it to compare the models on an equitable basis in terms of number of event points. The model has 160 binaries, 1,398 continuous variables and 2,559 constraints. Optimality is achieved in 1.518 CPUs after exploring 293 nodes. In comparison to model M, model M^{fix} solves to optimality in 133 less nodes and a slight reduction in the computation time. This result is non-intuitive as model M^{fix} has additional binary variables and constraints.

D.1.3 Case Study 3 - Flexible Maintenance Timing

A reduction of plant profitability due to planned maintenance events is observed in the above case study. It would be advantageous if the maintenance events could be carried out at a time when the units are not needed for critical processing tasks. To examine these potential benefits case study 2 is re-optimized with 2 hours of forward timing slack ($\eta_c = 2$) on both maintenance tasks. The problem was solved using model M^{flex}, which had a LP-relaxation of \$18,423. The optimal solution is determined as \$13,000 and is obtained with 9 event points in 2.29 CPUs after exploring 403 nodes. The optimal solution is depicted in Figure D.4. At 9 event points model M^{flex} produces 234 binary variables, 1,876 continuous variables and 3,386 constraints. It is interesting to note that at the same number of event points model M^{flex} outperforms model M. Model M requires searching an additional 1,140 nodes at the expense of 3.87 CPUs; this result is non-intuitive as model M^{flex} has more binary variables and is larger in size.

Maintenance events CU2 and CU3 occur at $T_6 = 7$ and $T_7 = 8$, corresponding to an optimized forward timing slack of 2 hours for each maintenance event. Moving the maintenance event forward allows for additional tasks on units U2 and U3 to be completed before maintenance events are scheduled to occur. As in the first case study, 10 kg of P1 and 3 kg of P3 are sold to market. These gains in productivity even for this small example showcase the need of combined consideration of maintenance and scheduling in a comprehensive optimization framework.

Figure D.4: Gantt chart of optimal solution for case study 3



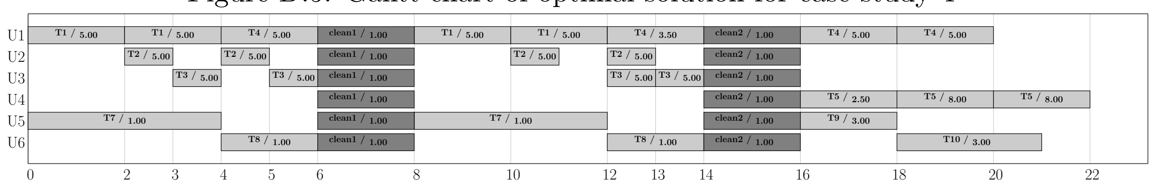
D.1.4 Case Study 4 - Recurring Maintenance Events

In particular industrial settings, maintenance may need to be carried out periodically. In this case study, the horizon of the problem is extended to 22 hours and a recurring process wide maintenance task, with $\Gamma = 7$ hours, is required to occur. Two maintenance events are planned to occur within the scheduling horizon. Maintenance event 1 is associated with task *clean1* ($\alpha_{clean1} = 2$) and maintenance event 2 is associated with task *clean2* ($\alpha_{clean2} = 2$). The maximum allowed timing of the maintenance events is given as $TC_1 = 7$ and $TC_2 = 16$.

The optimal solution is determined as \$21,500 and is obtained with 15 event points in 1,502 CPUs after exploring 92,047 nodes. At 15 event points, model M^{recur} has 390 discrete variable, 3,051 continuous variables, 5,553 constraints and an LP-relaxation of \$30,971.5. The optimal solution is shown as a Gantt chart in Figure D.5, where 18.5 kg of state P1 and 3 kg of state P3 are sold to market at the end of the horizon. The first and second maintenance events occur at $T_6 = 6$ and $T_{11} = 14$ hours, corresponding to an optimized 1 hour backward event timing on both maintenance events. The final maintenance event is completed at the 16^{th} hour of production, satisfying the final maintenance event timing constraints.

To examine the computational efficiency, core model M was re-solved with a time horizon of 22 hours and 15 event points. Model M has 300 discrete variables, 2,630 continuous variables and 4,764 constraints. After 7,200 CPUs a solution of $28,666 was found after exploring 369,473 nodes. An upper bound of $32,731 remained leaving an optimality gap of 14.18%. This observation is consistent with an earlier statement that 15 event points may be the upper limit on model complexity for the general global event continuous-time models [Méndez *et al.*, 2006]. In comparison, model M^{recur} resulted in significant computational savings as 277,426 fewer nodes were searched and optimality was achieved. This is an interesting result as the number of event points is at the supposed complexity limit but model M^{recur} solves to optimality in finite time. It is hypothesized the inclusion of the additional restrictions on processing units limits the number of available solutions, reducing the need to explore a large portion of the branch-and-bound tree.

Figure D.5: Gantt chart of optimal solution for case study 4



D.2 Conclusions

In this appendix a framework is proposed to incorporate maintenance events with fixed, flexible and recurring timing into the global event continuous-time scheduling model of Maravelias and Grossmann [2003b]. A well cited literature problem is modified and solved to exhibit the solution capabilities of the proposed formulation. It is seen that maintenance events of planned fixed timing can interrupt product pathways and significantly reduce plant profitability. It is shown that plant profitability can be improved by allowing the timing of the maintenance events to be optimized with production scheduling. In both case studies, a reduction in the nodes searched to optimality was observed in comparison to the base model with equivalent event points. The final case study considered the inclusion of a recurring maintenance event within an extended scheduling horizon. The optimal solution requires 15 event points and optimality is attained after 1,502 CPUs, in contrast to the base model which is left with a optimality gap of 14.18% after 7,200 CPUs. It is hypothesized that the inclusion of further restrictions on processing units reduces the number of available solutions and as such the brand-and-bound tree size.