MILP for Optimal Buffer Levels and Maintenance Scheduling

MILP Formulations for Optimal Steady-State Buffer Levels and Flexible Maintenance Scheduling

by

Kristin M. Davies, B.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 (2008)	McMaster University
(Chemical Engineering)	Hamilton, Ontario, Canada

TITLE:	MILP Formulations for Optimal Steady-State Buffer Levels and Flexible Maintenance Scheduling
AUTHOR:	Kristin M. Davies, B.Eng. (McMaster University, Canada)
SUPERVISOR:	Dr. C.L.E Swartz
NUMBER OF PAGES:	xiv, 119

ABSTRACT

In this thesis two industrially motivated problems, belonging to the same manufacturing process, are solved by mixed-integer linear programming (MILP) techniques. Initially, the problem of determining optimal steady-state buffer levels for minimizing unit failure impacts is examined. Next, a method for achieving an optimal and coordinated production and flexible maintenance schedule is studied under several different optimization goals. The process that is predominantly considered is a continuous processing plant that can be modelled as n units in series separated by $(n-1)$ buffer tanks.

Unit shutdowns due to equipment failure result in adverse economic consequences due to reduced production and costs associated with off-specification product. Mitigation of these effects may be possible if sufficient buffer capacity is available for parts of the plant to continue operation until the affected units are back in operation. A question that arises is what the optimal levels of the buffer storage units should be for use in normal operation, with insight as to what abnormal operating conditions may occur. This is a function of the expected unit failure frequencies and failure lengths, and the process dynamics. In this study, the problem is posed within a multi-period dynamic optimization framework. Historical records are used to determine key unit failure scenarios. The objective function considers the loss of profit associated with downtime, as well as a fixed cost of induced shutdowns due to buffers being either full or empty.

Production scheduling in coordination with maintenance scheduling is often completed simply by fixing maintenance and then scheduling production around the maintenance-associated process unit unavailability. While acknowledging that much research has been completed to determine optimal maintenance policies, allowing maintenance events to be scheduled with a small degree of flexibility may significantly improve the resulting makespan. Alternative formulations suited to batch processing and continuous processing are presented; where the key advantage to the latter formulation is the possibility of job-splitting. Flexible maintenance by means of a specified time interval, sequence-dependent cleaning, shared finite intermediate storage, and product deadlines are accounted for in this dynamic optimization problem. Considered objectives include: makespan minimization, throughput maximization, and intermediate inventory minimization.

Several case studies are provided for both problems, with the intention of demonstrating the functionality of the formulations as well as to indicate the possible process improvements upon implementation.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Christopher Swartz for his continued guidance and patience throughout the past two years; I would certainly not have attained this degree without his weekly meetings, input, and direction.

Thanks also go to the industrial partner, PepsiCo Foods Canada Inc., for graciously providing the key motivation for this thesis. Special thanks go to Diane Richard, Emily Nichols, Andrew Matchett, Kevin Bradley, and Jim Quigley.

In addition, I would like to thank my parents for their financial and emotional support throughout my time at McMaster. Above all, I would like to sincerely thank Darryl Wallace for his unwavering vote of confidence and continued motivation.

This thesis is dedicated to my grandfather, Vincent Davies, who thought that it was "quite sensible" for me to complete this degree, and who has been consistently inquiring about my schooling ever since kindergarten!

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, two industrially motivated problems (belonging to the same manufacturing process) are solved by mixed-integer linear programming techniques. Initially, the problem of determining optimal steady-state buffer (intermediate storage) levels for minimizing unit failure impacts is examined. Next, a method for achieving a coordinated production and flexible maintenance schedule is studied under several different optimization goals.

Process unit shutdowns within a multi-unit process commonly induce additional unit shutdowns, and furthermore, they invariably result in adverse economic consequences. Possibilities of such consequences include loss of production as well as the creation of off-specification product. Mitigation of these effects may be possible if sufficient buffer capacity is available to introduce a level of independence between the upstream and downstream processes, to allow the remainder of the plant to continue operation until the affected units are back in operation. Maximizing the potential of such a buffer storage, however, is not trivial. Relative equipment reliability must be considered when determining the optimal nominal buffer storage level. Although much research has been conducted to determine the size of intermediate storage tanks, little research has been performed to determine what level the tank should be kept at nominally.

The increasing demand for low-volume, high-value products in today's manufacturing industry has indirectly driven much research in the area of scheduling. The nature of these products has led to the creation of multi-product and multi-purpose processing plants, which subsequently leads to scheduling research. In order to maximize productivity and profitability, equipment and other resources must be used wisely so as to meet all production demands and minimize production costs. Although scheduling research is highly prevalent in literature, the level of coordination of production and maintenance scheduling has been sub-optimal; often production is simply scheduled around equipment unavailability that has been fixed by maintenance. Improved scheduling is feasible if maintenance events are not fixed, and rather scheduled with some amount of flexibility along with production jobs.

1.1 Motivation and Goals

The overall research goal is to apply multi-period dynamic optimization in coordination with mixed-integer linear programming to represent and solve a true industrial process problem. The objective, as formulated through collaboration with an industrial partner, is to minimize material losses caused by transitional periods (including unit failures, unit maintenance, and product changeovers). This objective is to be accomplished through the following three sub-objectives:

1. to determine the optimal intermediate storage levels;
2. to optimally schedule maintenance; and
3. to optimally schedule weekly production requirements.

The first sub-objective refers to the optimization of intermediate storage levels during normal operation with insight as to what abnormal operating conditions may occur. This is a function of the expected unit failure frequencies, failure duration, recovery time, and the process dynamics. Optimizing these levels will reduce plant-wide impacts caused by unit failures, hence saving material otherwise lost during additional shutdowns and subsequent start-ups.

The second and third sub-objectives are to be solved simultaneously in order to truly optimize the production line schedule. Opportunity for decreasing process upsets caused by maintenance is made possible by allowing maintenance events to be flexible by some small time interval. If job-splitting (forced maintenance before the current job is finished which then causes the same job to continue production at some point after the maintenance event finishes) can be avoided by starting maintenance either slightly earlier or later than initially desired, material that would otherwise be lost due to job-splitting (material purged during start-ups and shutdowns) is minimized, as is the total set-up or cleaning time.

The mathematical models, constructed to best represent true process characteristics, include both continuous and discrete decisions. Discrete decisions are required in the optimization problems, due to highly constrained processing rates (i.e. the triggering of a shutdown when throughput falls beneath a specified minimum value), the need to count shutdowns, and the product and maintenance scheduling tasks. Discrete decisions are represented in optimization problems as integer variables, and the presence of discrete decisions among continuous variables results in a mixed-integer linear programming problem; hence, mixed-integer linear programming and dynamic optimization are key features of the mathematical formulations developed for this problem.

1.1.1 Industrial Process: Ready-to-Eat Cereal

The industrial partner is a ready-to-eat (RTE) cereal manufacturer. As indicated by the name, ready-to-eat cereal requires no further cooking before consumption, in contrast to hot breakfast cereals, and it is generally made from one or more of the following main ingredients: corn, wheat, oats, and rice. The main ingredient is most often supplemented with fortifying ingredients as well as additional flavouring (Fast and Caldwell, 2000). A typical process flow diagram for an extruded ready-to-eat cereal line is provided as Figure 1.1.

Although several individual operations exist, the zero-wait nature of these continuous processes allows for the grouping of manufacturing and packaging activities, for mathematical modelling purposes, as shown in Figure 1.1. Preprocessing and mixing operations are generally completed in a batch nature, and are not considered to be a bottleneck in comparison with manufacturing and packaging, therefore in this thesis, "raw" materials or feeds are assumed to have completed the mixing stage, and their supply is unconstrained. The two processing units that are hence modelled throughout this thesis are manufacturing and packaging. The industrial partner also has an intermediate storage tank located between manufacturing and packaging - a key opportunity for optimization.

1.2 Main Contributions

The first contribution of this thesis is a mathematical formulation which allows for the optimization of intermediate buffer levels based on the relative reliability of individual process units. The reliability information (frequency and duration of failure) can easily be modified within the formulation framework to reflect any continuous production process of n units in series separated by $n-1$ intermediate finite storage

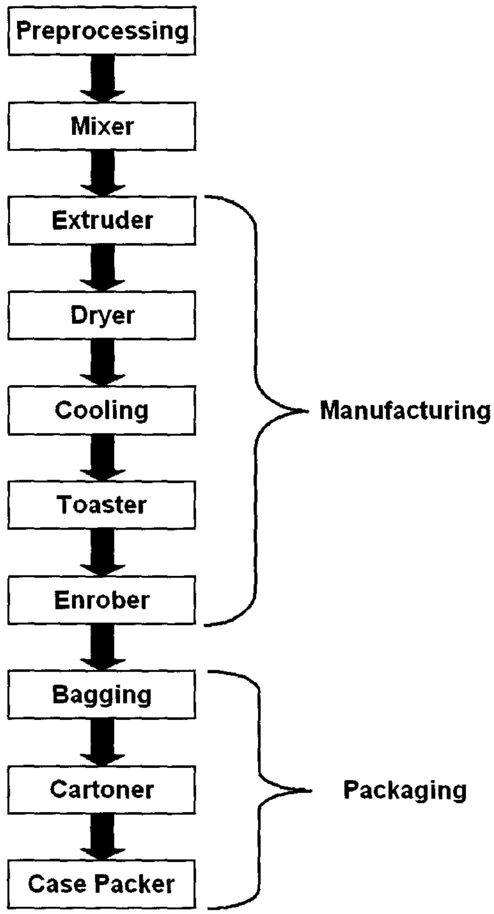

Figure 1.1: Typical extruded ready-to-eat cereal process, adapted from U.S.E.P.A. (1995), Lorenz and Kulp (1991), and Fast and Caldwell (2000).

tanks. This formulation was also shown to effectively handle process configurations including recycle loops.

The second contribution is a mathematical formulation which allows for optimal scheduling of p products through a multi-product plant (of n units in series separated by $n-1$ intermediate finite storage tanks) in addition to the scheduling of m required maintenance events. Product due dates, sequence-dependent cleaning, and hard maintenance constraints within a flexible maintenance "window" are accounted for. Alternative formulations are provided to more accurately represent either batch or continuous processes.

1.3 Thesis overview

Chapter 2 – Literature Review

The literature review covers four main topics: dynamic optimization; mixed-integer linear programming; steady-state buffer levels for failure uncertainty; and scheduling.

Chapter 3 – Steady-State Buffer Levels with Failure Uncertainty

Relative equipment reliability and shut-down characteristics have been considered as the major factors which determine the optimal nominal buffer storage level. Various key failure "scenarios" are simultaneously considered in order to achieve one overall optimal buffer level common to all scenarios. The "scenario" information includes details on which unit has failed, the downtime and recovery time, and frequency of failure. The failure scenario information can easily be altered to incorporate historical records provided by any qualifying process.

The optimization problem is then expanded to consider the possibility of sending intermediate product to a purge stream in order to avoid an induced unit shutdown when economically logical. In addition, the described objectives and problem formulation have been applied to a typical pulp mill process which includes a recycle stream.

Chapter 4 – Flexible Maintenance Scheduling

Optimal schedules over a given time horizon encompassing both production jobs and maintenance events are the outcome of the mathematical formulation of this chapter. The dynamic nature of this problem dictates that intermediate storage levels must also be modelled, especially since intermediate storage is finite and shared by all products. Additional complicating factors incorporated to reflect reality include the existence of sequence-dependent set-up times or cleaning between products, and product deadlines. Alternative formulations are presented to best suit batch or continuous processes.

Chapter 5 – Conclusions and Recommendations

A summary of this thesis is presented in the final chapter, which highlights the major results. Avenues for future work are also discussed here.

Chapter 2

Literature Review

This chapter is intended as a brief review of research topics relevant to this thesis. It is not intended as a comprehensive report on these topics; for more information, the reader is encouraged to consult directly with the primary references. The relevant research topics have been determined as: dynamic optimization, mixed-integer linear programming, steady-state buffer levels for failure uncertainty, and scheduling. Accordingly, these four topics will now be reviewed in this order.

2.1 Dynamic Optimization

The following section will introduce the reader to the topic of dynamic optimization. Background definitions and solution methods will be addressed. This is an important topic to be examined since all formulations presented in this thesis are prepared in this format.

2.1.1 Definitions

Differential Algebraic Equations (DAEs) are often used to describe the dynamic behaviour of a system. Typical examples of chemical process dynamics that are modelled in this fashion are mass and energy balances of systems in transition.

Dynamic Optimization (DO) The use of differential algebraic equations as constraints to an objective function forms what is commonly referred to as a dynamic optimization problem. The general form of a dynamic optimization problem, as defined by Cervantes and Biegler (2001), is given in the equations below:

$$\min_{z(t),y(t),u(t),t_f,t_f,p} \phi\Big(z(t_f), y(t_f), u(t_f), p\Big) \tag{2.1}$$

s.t.

time-variant constraints:

$$\dot{z}(t) = f\Big(z(t), y(t), u(t), t, p\Big) \tag{2.2}$$

$$h\Big(z(t), y(t), u(t), t, p\Big) = 0 \tag{2.3}$$

$$g\Big(z(t), y(t), u(t), t, p\Big) \leq 0 \tag{2.4}$$

initial conditions:

$$z(0) = z_0 \tag{2.5}$$

point conditions:

$$h_P\Big(z(t_i), y(t_i), u(t_i), t_i, p\Big) = 0 \tag{2.6}$$

$$g_P\Big(z(t_i), y(t_i), u(t_i), t_i, p\Big) \leq 0 \tag{2.7}$$

bounds:

$$z_L \leq z(t) \leq z_U \tag{2.8}$$

$$y_L \leq y(t) \leq y_U \tag{2.9}$$

$$u_L \leq u(t) \leq u_U \tag{2.10}$$

$$p_L \leq p \leq p_U \tag{2.11}$$

$$(t_f)_L \leq t_f \leq (t_f)_U \tag{2.12}$$

In the above, ϕ = scalar objective function, $z(t)$ = differential state profile vector, $u(t)$ = control state profile vector, $y(t)$ = algebraic state profile vector, t = time, p = time-independent parameter vector, f = differential equation constraints, h = algebraic equation equality constraints, g = algebraic equation inequality constraints. Finally, the subscript L refers to lower boundary limits, while the subscript U refers to upper boundary limits.

2.1.2 Solution Methods

Dynamic optimization problems are typically solved either by the indirect (variational) approach or by a direct (discretization) approach.

The indirect method relies on Pontyragin's Maximum Principle to solve the first order optimality conditions. Difficulties arise with boundary conditions, such that a two-point boundary value problem is obtained. In addition, direct approaches are known to be more numerically stable, according to Cervantes and Biegler (2001).

Direct approaches involve transforming the continuous-time problem into a discrete-time problem. This discretization approach can be broken down further into two categories: sequential methods and simultaneous methods. The sequential method is also known as "partial discretization" since only the control profiles are discretized; whereas the simultaneous approach is alternately known as "full discretization" as both the state and control profiles are discretized (Cervantes and Biegler, 2001).

Both direct approaches can be solved by a nonlinear programming (NLP) technique. However, the difference between the two methods is evident by the size and stability of the problem and solution. Cervantes and Biegler (2001) have found that while the sequential method emerges with multiple smaller problems (which are then solved iteratively), the simultaneous method emerges with one larger but also more stable solution.

2.1.3 Orthogonal Collocation on Finite Elements

Orthogonal collocation on finite elements (OCFE) is a popular method for discretizing the states in a simultaneous solution approach. Orthogonal collocation on finite elements corresponds to an implicit Runge-Kutta method which implies high order accuracy and good stability properties.

The four main ideas of orthogonal collocation on finite elements have been summarized by Swartz (2006) as follows:

- The domain of the original differential algebraic equation system is subdivided into finite elements (nFE).
- Within each finite element, the solution to the problem is approximated by a low-order polynomial.

- Orthogonal collocation is applied within each finite element.
- Function continuity, and sometimes derivative continuity, is imposed across the element boundaries.

A diagram depicting the important features of orthogonal collocation on finite elements can be found as Figure 2.1. Initially, every differential and algebraic variable is approximated by a polynomial. This is often accomplished with Lagrange interpolation polynomials. As can be seen in Figure 2.1, the state profile has been segmented into five parts, known as finite elements. Vasantharajan and Biegler (1990) have found that once segmented, two to three collocation points are sufficient for each element. The collocation points are typically chosen as the roots of an orthogonal polynomial family. A popular choice for these orthogonal polynomials is the Jacobi family. At each collocation point, the residual, obtained by substituting the approximating function into the differential / algebraic equation, is forced to be equal to zero. Finally, function continuity between successive finite elements is imposed by setting the differential variable at the left boundary equal to the polynomial approximation of the previous finite element. Further information on this method is available from Cuthrell and Biegler (1987).

2.2 Mixed-Integer Linear Programming

The following section will introduce the reader to the topic of mixed-integer linear programming. Background definitions, application areas, and solution methods will be addressed. Mixed-integer linear programming is used throughout all mathematical formulations provided in this thesis. Integer variables are highly useful in various optimization applications, and are therefore present in both Chapters 3 and 4. To minimize problem complexity (and hence computation time), significant effort was

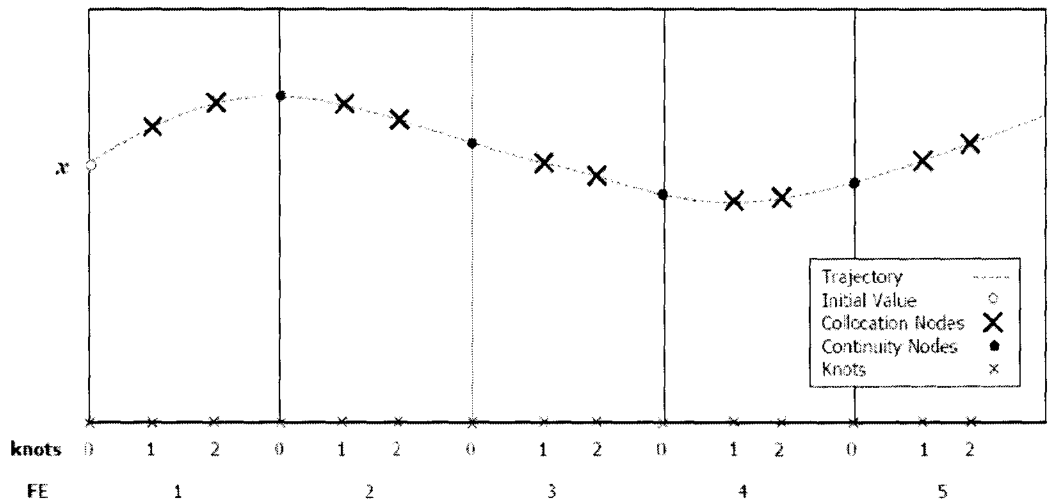

Figure 2.1: An example of orthogonal collocation on finite elements, as interpreted by Chong (2006).

put forth to keep all optimization formulations linear, due to the inherent presence of integer variables.

2.2.1 Definitions

Linear Programming - A linear program (LP) is defined as an optimization problem in which a linear objective function is maximized or minimized while satisfying a set of linear equality and/or inequality constraints (Winston, 1991). According to Gill *et al.* (1986), the general form of a linear program may be stated as follows, where x includes slack variables:

$$\min c^T x \tag{2.13}$$

$$\text{subject to} \qquad Ax = b, \qquad x \geq 0 \tag{2.14}$$

$$\text{where } x, c, b \in \Re^{\mathbf{n}} \quad \text{and } A \in \Re^{\mathbf{m \times n}}$$

Integer Programming - As defined by Winston (1991), an integer program (IP) is an linear program with the additional characteristic that all of the variables are nonnegative integers.

Mixed-Integer Linear Programming - The case of integer programs where only some of the variables are to be nonnegative integers (while the rest are continuous variables), results in what is known as a mixed-integer linear programming problem (MILP) (Winston, 1991). The general form of a mixed-integer linear program is given as follows, with x and y representing integer variables and continuous variables respectively.

$$\text{minimize} \quad c^T x + d^T y \tag{2.15}$$

$$\text{subject to} \quad Ax + By = b \tag{2.16}$$

$$x \geq 0 \tag{2.17}$$

$$y \geq 0 \tag{2.18}$$

$$\text{where } x \in \mathbf{Z}^{\mathbf{n_1}} \quad \text{and } y \in \Re^{\mathbf{n_2}}$$

2.2.2 The Branch and Bound Search

Integer and mixed-integer programs are commonly solved by a solution method known as the Branch and Bound method. According to Winston (1991), the first step of the Branch and Bound method involves solving the LP relaxation. A common LP relaxation is the problem obtained by treating the integer variables as continuous. By definition of the relaxed problem, the feasible region of any mixed-integer linear program is contained within the feasible region of the LP relaxation. Additionally, this implies that the optimal function value of the LP relaxation will be greater than or equal to the optimal value of the mixed-integer linear program function (if dealing with a maximization objective). Hence, the solution of the LP relaxation can be

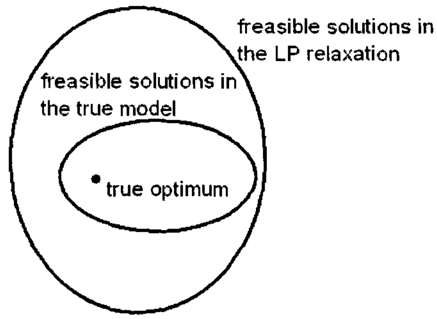

Figure 2.2: Location of true optimum in LP relaxation, as depicted by Rardin (1991)

regarded as a maximum or upper bound of the mixed-integer linear program. This concept is illustrated by Figure 2.2. In addition, this LP relaxation forms subproblem 0 of Figure 2.3.

If the optimal solution of the LP relaxation has all decision variables as an integer (or binary) value, then this is also the optimal solution of the mixed-integer linear program. Assume however, that this upper bound does not meet this criterion.

In order to find the optimal mixed-integer linear program solution, the feasible region of the LP relaxation must be partitioned. A relaxed integer variable that takes on a fractional value in the LP relaxation optimal solution (subproblem 0) is selected as a branching point. This variable is branched to the upper and lower integer value associated with the current fractional value; these integer values are included as upper and lower bounds respectively in the corresponding node subproblems. In Figure 2.3, x_2 has been selected as the branching point, which then creates subproblems 1 and 2.

Arbitrarily, one of subproblems 1 or 2 is selected as the next branching point. Clearly, the result of each branch is two new subproblems. At each new pair of subproblems, this procedure is repeated, selecting one variable to further branch based on a frac-

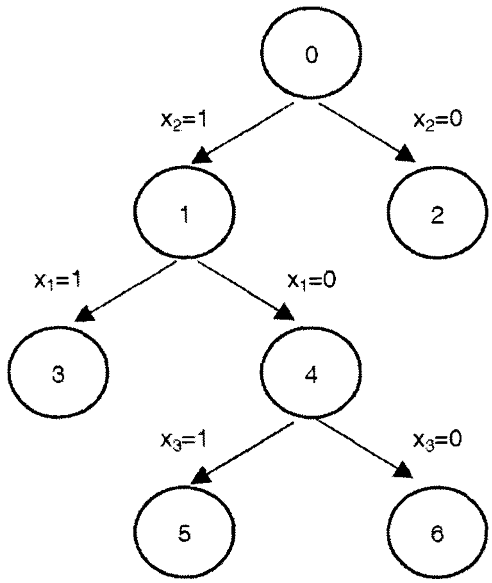

Figure 2.3: A binary search tree for the branch and bound method. (Note, arbitrary branching is shown here.)

tional decision variable. This process continues until one of three situations occurs:

- A *candidate solution* is reached. A candidate solution is one that has all (required) decision variables assigned integer values.
- The optimal function value of the subproblem does not exceed that of the current *lower bound*. A lower bound is the candidate solution with the highest objective function value that still remains less than or equal to that of the optimal LP relaxation solution (in the case of a maximization problem).
- The subproblem is infeasible (i.e. the constraints cannot be simultaneously satisfied).

Upon any of these three events occurring, no further branching is required from that node. Once the search-tree is complete, the optimal solution can easily be selected from the few candidate solutions (based on objective function value and feasibility) (Winston, 1991).

2.3 Steady-State Buffer Levels for Failure Uncertainty

The intention of installing intermediate storage (or buffer) tanks within a continuous process is to decouple the units directly upstream and downstream of the tank. Although much research has been undertaken to determine the optimal size of intermediate storage tanks, very little work has been done in regards to determining optimal nominal buffer storage levels. The relevant papers to this specific topic are now discussed.

Lee and Reklaitis (1989) used Fourier series expressions to develop a set of analytical expressions that calculate the minimum buffer capacity based on batch failure frequency and duration. A difference in this work is the restriction to batch processes; the only failure mode studied is an entire batch failure (i.e. all material in the batch is purged from the process). This restriction provides reason for Lee and Reklaitis (1989) to negate the effects of downstream failures when computing minimum buffer capacities. When studying continuous processes on the other hand (as is the case of Chapter 3 of this thesis), downstream equipment failures have a direct impact on the optimal upstream buffer level, as only a relatively small portion of material (the amount actually in the unit at the time of failure) is purged during a failure, and the remainder is forced to be held in the storage tank. The analysis of Lee and Reklaitis (1989) is also limited to two process units separated by a single storage vessel, and induced shutdowns are not considered.

A more recent paper, with some of the same authors as the Lee and Reklaitis (1989) work, is that of Huang *et al.* (2000). Here, a "fault accommodation" system is proposed which is based on dynamic optimization principles. The need for such a tool is stressed by the authors; however the main features of the tool are fault identification and subsequent recovery control profiles; optimal nominal buffer levels are not computed.

Optimal storage operating strategies were investigated by Dube (2000) with the goal of either minimizing abnormal operation time or minimizing shutdowns. Research goals included determining the longest possible shutdown with existing storage capacities and the plant-wide effect of such shutdowns, and determining conditions for which increased buffer capacity would increase overall plant production. Dube (2000) points to optimizing steady-state buffer levels based on shutdown probabilities as a key area for future work.

Balthazaar (2005) seemingly followed the recommendation of Dube (2000), and optimized both preemptive and reactive plant-wide responses to unit shutdowns. The emphasis of these responses was process flow rates, but he also briefly investigated nominal buffer tank levels as well. A two-tiered economic objective function was applied to a multi-period dynamic model to determine the buffer tank levels. Chong (2006) extended the optimal failure recovery work of Balthazaar (2005) by including uncertainty in the downtime estimates and process model. Here, no attempt is made at optimizing the nominal buffer levels, rather, the goal is to optimally utilize the amount of material initially in intermediate storage to buffer the remainder of the process during a shutdown. Chapter 3 of this thesis is a direct extension of the work both by Balthazaar (2005) and Chong (2006).

Another related research effort is that of Tolio *et al.* (2002). The goal of this work was to determine the average buffer level over a time horizon, but with no possibility for preventative or corrective action upon unit failures or shutdowns; in particular, no form of optimization is considered in their formulation. The model is an extension of the deterministic processing time model described in Germanin (1994), which itself was a modification of the models by Buzacott and Hanifin (1978) and Buzacott and Shanthikumar (1993). This work requires several restricting assumptions, and those that generate key differences to the work of this thesis are: i) that all machines have equal and constant processing times, ii) modelling is developed for discrete-part manufacturing, and iii) time is scaled so that the machine cycle takes one time unit. The average buffer level can be written as functions of various state probabilities, where the "state" of the system, s, refers to combination of the following characteristics at a given time t: the number of parts in the buffer at time t, the upstream failure mode at time t, and the downstream failure mode at time t. It is therefore essential for all probabilities to be calculated, which can be done as a Markov chain with M states. The Markov chain can be solved by solving a linear system of M equations in M unknowns, however, this becomes impractical with large N. Instead, Tolio *et al.* (2002) propose a methodology that only requires evaluation of the roots of an $(s+t)$ degree polynomial and solves a linear system of $(s+t-1)$ equations. This methodology first analyzes the Markov chain, and then makes a guess at the form that the internal states $(2 \leq n \leq N-2)$ assume. If this guess is correct, Tolio *et al.* (2002) maintain that it must then be possible to find a linear combination of these solutions that also satisfy the boundary conditions (and this is the solution of the Markov chain).

2.4 Scheduling

The existence of multiple-product production within a single plant has created the need for strategic scheduling of such plants. The following section will introduce the reader to the topic of scheduling in terms of key issues, general formulation approaches, and relevant research to the scheduling work of this thesis.

2.4.1 Introduction to Scheduling Issues

A need for scheduling in chemical process operations arises in multiple-product production within a single plant. Scheduling solutions generally aim to optimize both the sequence in which products will be produced as well as the appropriate start-up or completion time of each task. According to Mah (1991), a multi-product and multistage batch plant can be categorized as one of the following:

- *Multi-product plant or flowshop* - each product follows the same production path; uses the same equipment in the same order.
- *Multi-purpose plant or jobshop* - some products follow slightly different routes; they either have extra or deleted stages and/or they follow a slightly different equipment order.

Clearly, the type of plant has a direct impact on scheduling. Furthermore, the type and amount of intermediate storage within a batch process also has a direct impact on scheduling. The type and amount of intermediate storage can be classified into the four following self-explanatory categories (Mah, 1991):

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

Although the zero wait and no intermediate storage first appear to be similar concepts, they are in fact different. In no intermediate storage, the product material may reside within a processing unit for a period of time while waiting to enter the next unit. In zero wait however, the product material cannot be held in any processing unit once that stage of production is complete (Mah, 1991).

Common objective functions of a scheduling problem include the minimization of one or more of the following: makespan (completion time of the last product on last processing unit), total time required to complete the entire schedule, or job tardiness (how early or late jobs are completed).

The major factors that determine the optimal schedule include: the number of products to be scheduled, the number of processing units available, the selected objective function, processing times, precedence/precursor constraints, the type of intermediate storage available, and sequence-dependent cleaning / set-up times.

2.4.2 Formulation Methods

As summarized by MacRosty (2000), scheduling problems are typically formulated by one of three methods: the classical formulation, discrete-time formulation, or continuous-time formulation.

Classical formulations involve the use of allocation constraints and recursive relations to ensure physically feasible solutions and to calculate completion times respectively. It should be noted that the original formulation by Ku *et al.* (1987) is restricted to cases where unlimited intermediate storage holds. More recently, Mah (1991) developed a set of different recurrence relations which expands the model to cases of no intermediate storage.

As indicated by the name, discrete-time formulations are based on a discretized time horizon. The highest common factor of all processing times is often selected as the time interval. Once discretized, all processing events must lie within an interval. Shah *et al.* (1993) presented discrete-time formulations based on a state-task net-

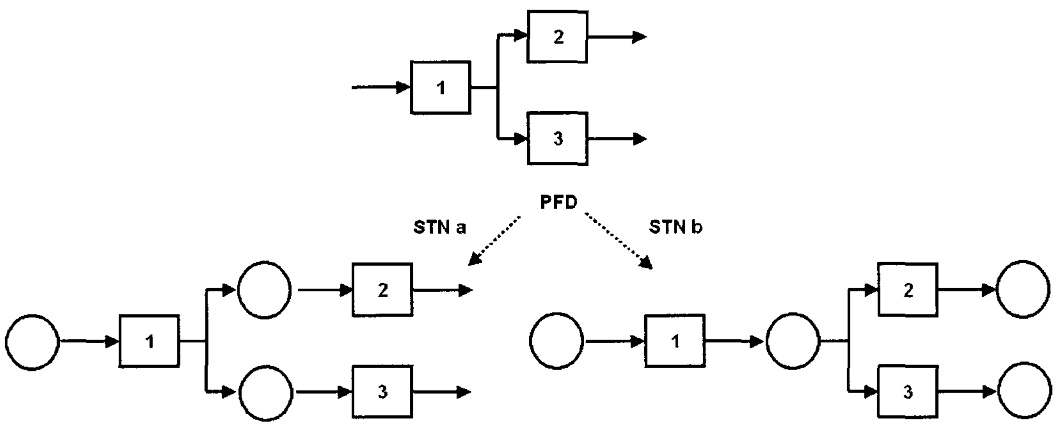

Figure 2.4: Two state-task network diagrams based on a single process flow diagram. Adapted from MacRosty (2000)

work (STN). State-task networks are an alternative to process flow diagrams, which attempt to describe the processes occurring rather than the physical pieces of equipment. Formulation of multi-product batch plant scheduling problems has certainly been simplified by incorporating state-task networks into the model. Figure 2.4 demonstrates this usefulness, as a single process flow diagram (PFD) can lead to two or more state-task networks. In STN a), the process flow diagram has been interpreted as process unit 1 creating two different materials, one that is subsequently processed by unit 2 and the other which is processed by unit 3. Alternatively, STN b) depicts a single output from processing unit 1 which can then be processed by either unit 2 or 3.

Similarly to the formulation by Shah *et al.* (1993), Ierapetritou and Floudas (1998) also present a formulation based on the state-task network approach. In contrast however, the Ierapetritou and Floudas (1998) formulation is based on a continuous-time model. Although continuous-time models have sometimes been found to show a larger integrality gap than its discrete-time counterpart, advantages include a smaller requirement of variables in order to obtain equivalent accuracy. The integrality gap is the difference between the relaxed problem objective function and the integral solution; large integrality gap worsens computational efficiency.

2.4.3 Flexible Maintenance Scheduling

Much integrated production and maintenance scheduling research is focussed on improving equipment availability by optimizing preventative maintenance. Preventative maintenance is not set by a pre-defined time-line, but rather by the increasing probability of failure, hence the completion of such a maintenance task resets the probability of failure which then improves the expected makespan. This research is not directly relevant to this thesis, and hence has not been summarized in this literature review. Several such probabilistic models exist, and include the recent work by Cassady and Kutanoglu (2005) and Amari (2006).

A more related research focus involves scheduling production jobs around equipment unavailability (i.e. when maintenance activities are pre-specified at a fixed time). Lee (1996) studied these fixed equipment unavailability jobs under both resumable and non-resumable production job constraints. Non-resumable indicates that if a job is not complete before the scheduled maintenance event, it must be restarted after the maintenance event. While the formulation by Lee (1996) could handle only one unavailability time period, Liao and Chen (2003) provided a formulation that could handle multiple fixed unavailability time periods.

Research on scheduling with flexible unavailability (maintenance) has been provided by Yang *et al.* (2002). This formulation could handle one unavailability time period, however production jobs were constrained to being non-resumable. Qi *et al.* (1999) and Chen (2008) both studied non-resumable production jobs as well, but were able to accommodate multiple flexible maintenance events. Qi *et al.* (1999) accommodate flexible maintenance by specifying two parameters in the model: maximum allowable continuous working time (T), and maintenance duration (t). The number of maintenance events is a decision variable, which is dependent on the total job processing time and the maximum allowable continuous working time. Maintenance events and production jobs are scheduled in conjunction with one another, always ensuring that a maintenance event takes place between every T time periods of continuous job processing. Chen (2008) accommodates flexible maintenance through a different approach - by specifying an acceptable time interval, $[u, v]$, over which a maintenance event of length w (where $w \leq v - u$) can and must occur. In doing so, multiple maintenance events with various characteristics ($[u, v], w$) can be scheduled in conjunction with the production jobs. A key restriction in this work however is that jobs are non-resumable.

Scheduling resumable production jobs in coordination with flexible maintenance is a research area that is yet to receive much attention. Graves and Lee (1999) and Brandolese *et al.* (1996) are two exceptions. In Brandolese *et al.* (1996), maintenance events are assigned an "optimal" or "desired" start time, which is associated with the smallest allocation cost. Maintenance events are allowed to be shifted away from this desired start time, or even bypassed completely, since no hard constraint exists to force maintenance to begin. As this event moves further from the desired start time, the allocation cost increases, which has a direct impact on the objective value, where objectives include minimization of total schedule cost, respecting due dates, and minimization of total plant utilization time. In addition, the number of maintenance events is initially unknown in the Brandolese *et al.* (1996) formulation, as it is dependent on the total processing time of an equipment unit. Graves and Lee (1999) successfully schedule production jobs in conjunction with a specified number of flexible maintenance events. Production jobs are considered "semi-resumable", meaning that if a job is interrupted for maintenance, this same job must be the first job that begins after maintenance is completed, and the setup time for this job must be repeated. Jobs cannot be interrupted by other jobs.

This work by Graves and Lee (1999) is most closely related to the scheduling work presented in this thesis; however key differences exist. The formulation by Graves and Lee (1999) includes a setup time for each job, however these set-up times are sequence-independent. This thesis provides formulation capable of sequence-dependent set-up times (or cleaning time), as well as fully resumable jobs. Jobs interrupted by maintenance may resume at any point later in the schedule, and jobs can be interrupted by other jobs, not only by maintenance. In addition, the Graves and Lee (1999) formulation, as with all other formulations reviewed here, is limited to scheduling of one machine in series, whereas this thesis presents a more general formulation which includes the capability of handling n units in series.

The formulation of scheduling problems in this thesis (Chapter 4) is based on the work by Kondili *et al.* (1993). This paper presents a general mixed-integer linear program formulation capable of optimizing short-term scheduling of batch operations, and it is based on a discrete time, state-task network approach. "Short-term" scheduling refers to the fact that the schedule is driven by product orders, and hence, the objective function maximizes profit through the value of its end products. Numerous constraints are provided to handle allocation constraints, finite capacity limitations, material balances, product due dates and raw material receipts, temporary unavailability of equipment, limited utilities and manpower availability, sequence-dependent and / or frequency-dependent cleaning, material storage within equipment, continuous feed and product approximations, multi-purpose storage, and limited equipment connectivity. Several of the key constraints are directly used in Chapter 4, while others are modified, some are omitted, and new constraints are included. It is noted in the concluding remarks of the Kondili *et al.* (1993) paper that the drawback to this formulation is the resulting mixed-integer linear program size, which in turn impacts the computation time and / or expense. Section 4.6 of this thesis discusses some of these associated computational issues.

Another related research effort based on the Kondili *et al.* (1993) state-task network approach is the work by Dedopoulos and Shah (1995). Here, flexible maintenance and non-resumable production are optimally scheduled for multipurpose plants within a mixed-integer linear programming structure. Every maintenance task is assigned an earliest possible start time as well as a latest possible completion time. A key difference in this work is the optimization objective, which is to maximize the "value-added" which is a measure of profitability defined as the difference between inventory values at the end and beginning of the time horizon minus the storage costs, value of material received from external resources, value of deliveries, cost of utilities consumed, and maintenance costs. The maintenance costs are broken into three sub-costs related to: the frequency and time that maintenance crew is called in, the time at which the equipment unit is maintained, and which crew is used. Maintenance resources (the availability of various crews, the maintenance cost associated with each crew, etc) are a major focus of this work. Additionally, this formulation is restricted to batch processing due to the absence of any form of material balance.

It is worth noting that the recent in-depth review paper (over 110 references) by Mendez *et al.* (2006), which is aimed at summarizing main features, strengths, and limitations of existing short-term batch scheduling modelling and optimization techniques, makes only a single reference to maintenance scheduling. The single mention of maintenance scheduling is in regards to problem classifications, where maintenance is described as a type of resource constraint; no indication of maintenance as a scheduling decision exists. Similarly, Floudas and Lin (2005) wrote a review paper on the mixed-integer linear programming advances for chemical process systems scheduling that references 80 papers, yet the only mention of maintenance scheduling is the work by Dedopoulos and Shah (1995).

Pistikopoulos has been involved in a number of research papers that account for maintenance considerations at the process design stage. These papers include Vassiliadis and Pistikopoulos (1998a), Vassiliadis and Pistikopoulos (1998b), Vassiliadis and Pistikopoulos (1999), Thomaidis and Pistikopoulos (1995a), Pistikopoulos *et al.* (2000), Thomaidis and Pistikopoulos (1994), Pistikopoulos *et al.* (1996), and Thomaidis and Pistikopoulos (1995b). These papers generally focus on integrating maintenance policies in the optimal life-cycle design of a plant. An expected profit objective function is used to quantify the effects of various maintenance policies and other uncertainties. In addition, Pistikopoulos *et al.* (2001), Vassiliadis and Pistikopoulos (2001) and Vassiliadis *et al.* (2000) have considered coordination of production and maintenance scheduling by defining a "system effectiveness" objective function that balances increased profit from increased equipment availability due to maintenance, and increased maintenance costs through mixed-integer linear programming. The focus of these papers is to determine the interactions between process design parameters and production and maintenance planning, hence a long time horizon of two years is considered in both examples presented. A multi-purpose plant which allows for rerouting of production to minimize adverse effects associated with equipment failures is considered.

Table 2.1 has been included to summarize some of the key differences between the discussed related research efforts and the present study included in this thesis.

Key Difference	Related Research	Present Study
fixed equipment unavailability	Lee (1996),Liao and Chen (2003), Kondili *et al.* (1993)	flexible equipment unavailability
non-resumable (batch) or semi-resumable production jobs	Qi *et al.* (1999),Chen (2008), Lee (1996),Yang *et al.* (2002), Graves and Lee (1999), Dedopoulos and Shah (1995)	fully-resumable production jobs
No. of maintenance events is a decision variable	Brandolese *et al.* (1996)	No. of maintenance events specified as a parameter
Soft maintenance constraints and maintenance can be cancelled	Brandolese *et al.* (1996)	Hard maintenance constraints and maintenance cannot be cancelled
obj: max "value-added"	Dedopoulos and Shah (1995)	several objective functions including min "makespan"
flexible equipment sizes (design problem)	Pistikopoulos *et al.* (2001), Vassiliadis *et al.* (2000), Vassiliadis and Pistikopoulos (2001)	fixed equipment sizes (operation of existing plant problem)

Table 2.1: Key differences in related research efforts.

Chapter 3

Steady-State Buffer Levels with Failure Uncertainty

This chapter introduces the topic of buffer level optimization, then presents a mixed-integer linear programming formulation, followed by an examination and discussion of various case studies, and finishes with a chapter summary. It should be noted that all case studies in this chapter were implemented in AMPL, and solved by CPLEX.

3.1 Introduction

The purpose of this study is to determine the optimal nominal steady-state level of an intermediate storage unit (buffer) for use in normal and continuous operation, with insight as to what abnormal operating conditions may occur. This is a function of the expected unit failure frequencies and failure lengths, and the process dynamics. In this work, the problem is posed within a multi-period dynamic optimization framework.

Although much research has been conducted to determine the size of intermediate storage tanks, little research has been performed to determine what level the tank should be kept at nominally, hence the motivation to do so in this chapter.

The term "failure scenario" is used in this chapter to refer to the location of the failure, as well as the frequency and length of the failure. For example, in this chapter when studying a 2-unit process separated by 1 buffer, two or more "failure scenarios" are considered. For the case of two failure scenarios, the first failure scenario has been selected as an upstream unit failure while the second is a downstream unit failure. Alternatively, the two failure scenarios could have been chosen to both be upstream failures of different failure durations, or two downstream failures of different failure durations. Selection of the failure scenarios should be done to reflect the most prominent failures that occur in a given process. The general process configuration studied here, as well as some nomenclature are illustrated in Figure 3.1.

As illustrated by Figure 3.1, $F_{u,s}[t]$ refers to the flow rate (kg/min) of process unit u in scenario s at time interval t. Similarly, $B_{u,s}[t]$ refers to the contents (kg) of buffer unit u in scenario s at time interval t. In all cases, $s = 1, .., S$, and $t = 1, .., T$. In reference to flow rates $u = 1, .., U$, while in reference to buffer units $u = 1, .., U - 1$. This is due to the characteristic of the process being studied, in which $U - 1$ buffers are used to separate U units.

As mentioned, Figure 3.1 b) and Figure 3.1 c) depict only two failure scenarios; many additional failure scenarios could be considered to account for different failure scenarios on the same unit. For example, the upstream unit (or unit 1) could fail from either electrical or mechanical disturbances. The expected failure frequency for each of these failure modes will be different, as will the failure length. In this work, we first present only two failure scenarios (one scenario per processing unit), however due to the multi-period approach, additional failure scenarios can easily be handled. A later case study incorporates ten scenarios, to reflect five possible failure modes for each of 2 processing units. Also, the methods and results presented here are for

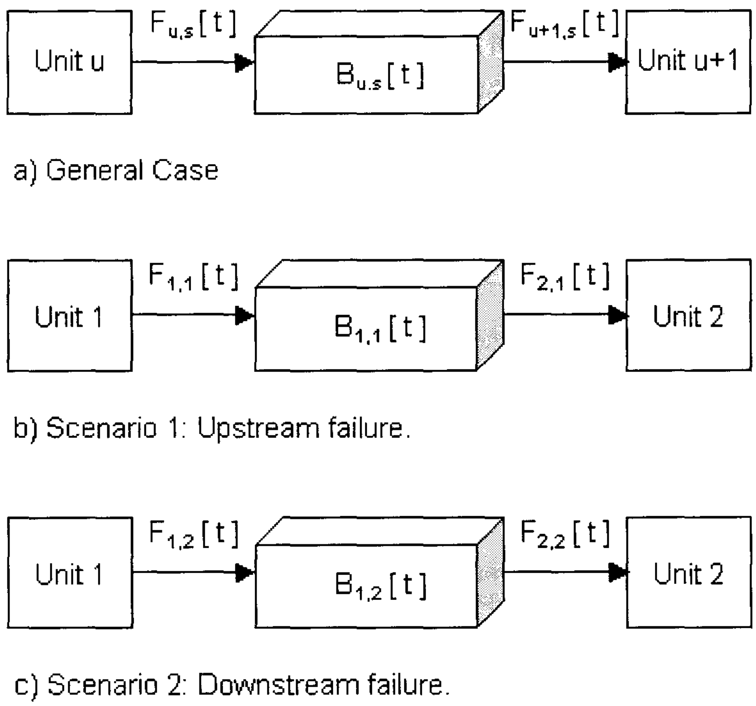

Figure 3.1: Depiction of 2-unit 1-buffer production system and associated 2 failure scenarios.

simple 2-unit and 3-unit systems, but can easily be extended to any n-unit process separated by $n-1$ buffers.

3.2 Formulation

It is important to understand some key terminology before examining the problem formulation. Figure 3.2 demonstrates two defined time periods for any given shutdown. These times will be used in the problem formulation, as well as in the results discussion. The first time period is called "shutdown duration". As illustrated, this is the time in which the failed unit has a flow rate of 0 kg/min. Next, "restoration time" is the period in which the failed unit is again operable. At the end of this time period, both the failed unit and the remaining units must be back to their original flow rates.

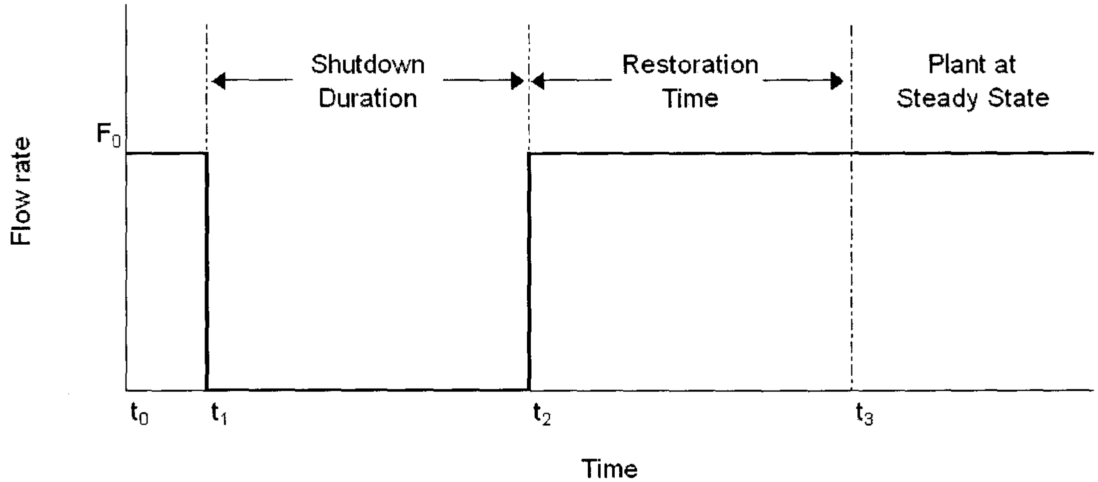

Figure 3.2: Depiction of important shutdown terminology.

The objective function has been formulated to optimize overall plant economics - both revenue and costs are taken into account:

$$\max_{F_{U,s},B_{u,0}} \sum_{s=1}^{S} f_s\Big(\sum_{t=0}^{T} R_U F_{U,s}(t)\Delta t - \sum_{u=1}^{U} N_{u,s}C_u\Big). \tag{3.1}$$

Revenue has been incorporated by correlating it with the end product. The furthest downstream flow rate in each scenario ($F_{U,s}$) is summed over the horizon and multiplied by the time interval, Δt, as well as by a constant factor, R_U. The result is a dollar value. Maximization of this term ($F_{U,s}\Delta t R_U$) over time hence correlates to a maximization of revenue. Additionally, shutdown costs are included by counting the number of unit shutdowns that occur ($N_{u,s}$) in each scenario, and multiplying them by a fixed cost factor specific to each unit (C_u) and relative to revenue associated with the end product. It should be noted that shutdown costs are already inherent in the objective function (as shutdowns clearly affect the maximization of profit); however, an additional shutdown fixed cost has been included to increase the significance of a failure or shutdown event. Finally, the relative unit failure frequencies are used to weight the importance of each scenario (f_s).

The initial conditions stated below pertain to the buffer level and flow rates, where $B_{u,0}$ is a decision variable corresponding to the optimal initial buffer level consistent with all failure scenarios for buffer unit u and $F_{u,0}$ is the initial flow rate consistent with all failure scenarios for process unit u.

Initial Conditions:

$$B_{u,s}(0) = B_{u,0} \tag{3.2}$$

$$F_{u,s}(0) = F_{u,0} \tag{3.3}$$

Next, dynamic path constraints are enforced throughout the entire time horizon.

Dynamic Constraints:

$$B_{u,min} \leq B_{u,s}(t) \leq B_{u,max} \tag{3.4}$$

$$B_{u,s}(t+1) - B_{u,s}(t) = \Delta t[F_{u-1,s}(t) - F_{u,s}(t)] \tag{3.5}$$

Equation 3.4 ensures that the buffer level remains within its physical limits, while equation 3.5 is a mass balance which calculates the buffer level change based on the difference in upstream and downstream flow rates. Note that the flow rates are assumed to be constant over each time interval.

Integer constraints are required to count the number of shutdowns that occur. The variable $w_{u,s}(t)$ is a binary variable that indicates whether a process unit is online ($w_{u,s}(t) = 1$) or offline ($w_{u,s}(t) = 0$). The equations below are used to determine $N_{u,s}$, the number of shutdowns of unit u in scenario s.

Integer Constraints:

$$-a_{u,s}(t) \leq w_{u,s}(t+1) - w_{u,s}(t) \leq a_{u,s}(t) \tag{3.6}$$

$$N_{u,s} = 0.5 \sum_{t=0}^{T-1} a_{u,s}(t) N_{u,s} = 0.5 \tag{3.7}$$

$$a_{u,s}(t) \geq 0 \tag{3.8}$$

Equation 3.6 is used to determine the absolute value of the difference between consecutive indicator variables, and since the effect of the objective function is to minimize the number of shutdowns, the bounds on equation 3.6 are tight and $a(t)$ is inherently binary in value. Finally, the variable $N_{u,s}$ is the calculated number of shutdowns via equation 3.7. We note that the units are assumed to be initially operational, and that normal operation is resumed by the end of the restoration period, as shown in Figure 3.2, thus $w_{u,s}(0) = w_{u,s}(T) = 1$.

The flow rate associated with the unit which undergoes a failure in any given failure scenario is denoted by $F_{f,s}$, and the flows corresponding to the remaining online units are denoted by $F_{f',s}$. The shutdown condition for online units has been specifically formulated to handle the discontinuous flow rate logical condition of $F_{u,min} \leq F_{u,s} \leq F_{u,max}$ or else $F_{u,s} = 0$. The significance of this construct is discussed in Section 3.2.1.

Shutdown Conditions:

$$F_{f,s}(t) = 0, \qquad t_1 \leq t < t_2 \tag{3.9}$$

$$w_{u,s}(t) F_{u,min} \leq F_{f',s}(t) \leq w_{u,s}(t) F_{u,max}, \qquad t_1 \leq t < t_2 \tag{3.10}$$

During restoration, all units (including that which had failed) are again operable, and therefore equation 3.10 must apply to all units,

Restoration Conditions:

$$w_{u,s}(t) F_{u,min} \leq F_{u,s}(t) \leq w_{u,s}(t) F_{u,max}, \qquad t_2 \leq t \leq t_3. \tag{3.11}$$

Note that equation 3.11 holds for $t_1 \leq t \leq t_2$ as well as for $t_2 \leq t \leq t_3$ for all non-failing units ($F_{f',s}$).

Finally, at the end of the optimization horizon, steady-state must be resumed; the buffer levels must return to their original values and all process units must be back the their original flow rates ($F_{u,0}$). These conditions are enforced by the following:

End-point Constraints:

$$B_{u,s}(T) = B_{u,0} \tag{3.12}$$

$$F_{u,s}(T) = F_{u,0} \tag{3.13}$$

3.2.1 Handling Flow Rate Discontinuities

The importance of equations 3.10 and 3.11 is to reflect realistic manufacturing conditions. Suppose that a unit shutdown is deemed to have occured when the flow associated with this unit is reduced to zero. The optimization would in this case tend to reduce the flow to an infinitesimally small amount, in order to avoid an induced shutdown to which a fixed cost, C_u has been associated. This is clearly unrealistic. Equation 3.10 forces the unit to shut down if its associated flow is reduced below a specified minimum operational limit, $F_{u,min}$.

To further explain this issue, consider an example of a 2-unit 1-buffer system. Suppose the upstream unit undergoes an induced failure. According to this specific failure scenario, the upstream failure lasts long enough that all intermediate products that exist in the buffer are depleted, and hence causes an additional downstream shutdown. This additional shutdown incurs a large fixed cost to the objective function. In order to avoid such a cost (and without equations 3.10 and 3.11 in place), the flow rate of the downstream unit can instead, simply be lowered to a minuscule value, which does not deplete a significant amount of material from the buffer before the upstream unit has recovered. In reality, however, many limitations (mechanical, chemical, etc.) exist which prevent a manufacturing process from operating at such low production rates. It is for this reason that equations 3.10 and 3.11 are in place.

A second related reason for implementing these discontinuous flow rate conditions is to reflect realistic start-up conditions. Commonly, the initial material created at low flow rates in a given process will not meet manufacturer specifications, and hence is purged from the process even before entering intermediate storage. Returning to the above example, once the upstream unit has recovered, it must begin production at or above the specified minimum flow rate (pertaining to the minimum acceptable product quality conditions) when equations 3.10 and 3.11 are implemented.

3.3 Case Studies

Although many parameters and case studies have been examined, only selected results are given here. A very basic system is examined first to gain a thorough understanding of the process characteristics, and then more complicated cases which generate less intuitive results follow.

3.3.1 2-Unit 1-Buffer System

In this simple 2-unit 1-buffer system, the effect of expected unit failure frequencies is examined. Two failure scenarios are considered simultaneously - one upstream and one downstream failure. Table 3.1 displays the pertinent case study parameters, while Table 3.2 displays the case study results. All variables are self-explanatory or have previously been discussed.

In Table 3.2, results are first presented for the case of unit 1 being less reliable (or

Parameter	Value	Units
$B_{1,min}$	0	kg
$B_{1,max}$	100	kg
Δt	1	min
T	60	min
$F_{u,min}$	10	kg/min
$F_{u,max}$	18	kg/min
$F_{u,0}$	15	kg/min
C_u	2000	$
R_U	1	$/kg
restoration	10	min

Table 3.1: 2-unit 1-buffer case study parameters.

Failure Duration	Optimal Buffer Level (kg)		
(min)	$f_1 = 0.8, f_2 = 0.2$	$f_1 = f_2 = 0.5$	$f_1 = 0.2, f_2 = 0.8$
3	30 - 70	30 - 70	30 - 70
4	40 - 60	40 - 60	40 - 60
5	50	50	50
6	60 - 100	0 - 40 , 60 - 100	0 - 40
7	70 - 100	0 - 30 , 70 - 100	0 - 30
8	80 - 100	0 - 20 , 80 - 100	0 - 20
9	0 - 100	0 - 100	0 - 100
10	0 - 100	0 - 100	0 - 100
11	0 - 100	0 - 100	0 - 100
12	0 - 100	0 - 100	0 - 100

Table 3.2: 2-unit 1-buffer case study results.

more likely to fail) than unit 2, with unit 1 failing 80% of the time. The next column gives the optimal buffer regions for equal unit failure likelihood, and finally, results are shown for when unit 2 is less reliable (failing 80% of the time). In all three cases, the results in Table 3.2 are sectioned into 3 groups: failure durations between 3-5 minutes, 6-8 minutes, and 9-12 minutes. The first and third result groups share equal optimal regions regardless of failure frequencies, while the second group shows different results for all three cases.

The first result group (failure durations between 3-5 minutes) indicates that the optimal buffer level is centrally located around the 50% buffer level (which in this case is 50 kg). Since both upstream and downstream failure durations are quite short, a central initial buffer level can withstand either failure without forcing an additional shutdown. (i.e. The upstream failure is short enough that the buffer will not be fully depleted, and the downstream failure is short enough that the buffer will not be fully accumulated.) As failure duration increases, however, the optimal region shrinks, and eventually becomes a single unique solution. Clearly, the longer the failure, the higher the minimum buffer level should be (to accommodate more depletion associated with the upstream failure) and the lower the maximum buffer level should be (to accommodate more accumulation associated with the downstream failure).

The specific values of the minimum and maximum optimal levels are directly related to the failure duration and minimum process unit flow rate. For example, a failure duration of 3 minutes results in an optimal region of 30 kg to 70 kg. During the upstream failure scenario, the downstream unit can lower its flow rate to the minimum allowable value of 10 kg/min (in order to reduce the failure impact, but without incurring a fixed shutdown cost). The effect on the buffer is then dictated by the difference in upstream and downstream flow rates over this 3 minute shutdown duration. As such, 30 kg are depleted from the buffer during the failure duration $\left((3min)(0 - 10kg/min) = -30kg\right)$. Therefore, 30 kg is the minimum initial buffer level that will result in an optimal solution. Similarly, during the downstream failure scenario with the upstream unit lowered to its minimum flow rate of 10 kg/min, a total of 30 kg will be accumulated in the buffer during the failure duration. For this reason, 70 kg is the maximum initial buffer level that will result in an optimal solution. Hence, in this result group, the optimal buffer level region can be calculated using the formula below. Note that t_{FD} refers to failure duration (min), and B_{opt} refers to the optimal nominal steady-state buffer level (kg).

$$t_{FD}F_{u,min} \leq B_{opt} \leq B_{u,max} - t_{FD}F_{u,min} \tag{3.14}$$

The above formula was determined upon examination of the optimization results. In order to initially obtain these "optimal regions" through dynamic optimization, a multi-tier approach was required. In the first tier, the plant economics were maximized, using the objective function of equation 3.1. Then in the second tier, the buffer level was maximized (or minimized) while still maintaining the first tier objective value. A similar but more extended discussion of this approach is continued for the 3-unit case study in Section 3.3.3

The third result group (failure durations between 9-12 minutes) reveals total non-uniqueness. The same objective value is attained from any initial buffer level, regardless of the failure frequencies. This result is easily explained. Due to the long failure duration, both failure scenarios will cause an additional shutdown, regardless of the initial buffer level. For failure durations of 11 and 12 minutes, the buffer will be fully accumulated or depleted before the failure duration finishes, hence causing an additional shutdown. For failure durations of 9 and 10 minutes, although the buffer may not be fully accumulated or depleted, an additional shutdown will occur in order to fulfill the end-point constraints of equation 3.12 and 3.13. This will be further explained by the discussion of the results in Table 3.4.

The second result group (failure durations between 6-8 minutes) reveals the impact of failure frequency on the optimal buffer level. When the upstream unit is more likely to fail, the optimal region is located within the upper buffer capacity. Conversely, when the downstream unit is more likely to fail, the optimal region is located within the lower buffer capacity. Finally, equal unit reliability results in a union of the two previously discussed cases. The upper capacity buffer level associated with upstream unreliability ensures that enough material is in the buffer to sustain an upstream failure without causing an additional downstream shutdown. Of course it is also possible that the downstream unit will fail, in which case, the buffer will quickly fill to capacity and force an additional downstream failure; however, this situation is less likely. The same, but opposite explanation can be applied to the lower capacity buffer level associated with downstream unreliability. The split optimal buffer level region obtained with equal unit reliability is easily explained: if the initial buffer level is in the central buffer capacity, both an upstream failure and a downstream failure will cause additional shutdowns. If kept in the upper buffer capacity, however, only a downstream failure would cause an additional shutdown; there would be enough material in the buffer to sustain an upstream failure. Similarly, if kept in the lower buffer capacity, only an upstream failure would cause an additional shutdown; there would be enough free space in the buffer to sustain a downstream failure. Therefore, it is optimal for the level to be within the upper capacity or lower capacity, but sub-optimal to be centrally-located.

As mentioned, the key the parameters that determine the optimal buffer limits are failure duration and minimum flow rate. In the case of an unreliable upstream unit with a failure duration of 6 minutes and with the downstream unit flow rate reduced to 10 kg/min (to minimize impact), a total of 60 kg will be depleted from the buffer; hence, the buffer level should be kept at or above 60 kg. In the case of an unreliable downstream unit with a failure duration of 6 minutes and with the upstream unit flow rate reduced to 10 kg/min, a total of 60 kg will be accumulated in the buffer; hence, the buffer level should be kept at or below 40 kg.

These observations lead to analytic expressions for the optimal initial buffer levels for the second result group (failure durations between 6-8 minutes). The calculation of the optimal buffer region is clearly dependent on the failure frequencies. For $f_1 > f_2$, the optimal region can be calculated by

$$t_{FD}F_{u,min} \leq B_{opt} \leq B_{u,max}. \tag{3.15}$$

For $f_1 < f_2$, the optimal region should be calculated by

$$B_{u,min} \leq B_{opt} \leq B_{u,max} - t_{FD}F_{u,min}. \tag{3.16}$$

Finally, for $f_1 = f_2$, the optimal region is determined by

$$B_{u,min} \leq B_{opt} \leq B_{u,max} - t_{FD}F_{u,min} \tag{3.17}$$

and

$$t_{FD}F_{u,min} \leq B_{opt} \leq B_{u,max}. \tag{3.18}$$

The last point to be examined in this case study is why the three result groups are split apart at certain locations. Why does the trend of centrally located optimal regions switch between failure durations of 5 and 6 minutes? And why does optimal region switch again between 8 and 9 minutes?

The switch between the first and second result group is simple. The failure durations change from being short enough that all additional shutdowns can be avoided (by slowing the other operational unit to 10 kg/min), to long enough that at least one additional shutdown is inevitable (even when slowing the other operational unit).

The impact-minimizing step of reducing the other operational unit to the lowest possible flow rate during a failure is illustrated in Table 3.3. A 4 minute failure of unit 1 will result in a 40 kg depletion of the buffer, rather than a 60 kg depletion, when setting $F_2 = 10$ kg/min as opposed to leaving it at 15 kg/min. A similar impact reduction is noted for a failure of unit 2. For a 6 minute failure, the minimal depletion / accumulation is 60 kg. Since this value represents more than 50% of the buffer capacity (which is 100 kg in this case), the optimal buffer level region switches from being centrally located to being either upper, lower, or split (depending on unit failure frequencies).

t_{FD} (min)	F_1 (kg/min)	F_2 (kg/min)	Depletion / Accumulation (kg)
4	0	15	-60
4	0	10	**-40**
4	15	0	+60
4	10	0	**+40**
6	0	15	-90
6	0	10	**-60**
6	15	0	+90
6	10	0	**+60**

Table 3.3: Failure impact minimization and switching from result group 1 to 2.

The switch between the second and third result groups is no longer impacted by only the failure duration and minimum flow rate. Here, the restoration length is also a key parameter. In this case study, the restoration time is 10 minutes. The goal of the restoration period is to return the buffer level to its original position. In the case of an upstream failure scenario, it is then necessary to allow $F_1 > F_2$ in order to re-fill the buffer. Keeping in mind that maximizing F_2 is one of the goals of the objective function, then Table 3.4 displays some of the combinations of flow rates that can occur during this 10 minute restoration (t_R).

t_{FD} (min)	min failure dep/acc (kg)	t_R (min)	restoration F_1 (kg/min)	restoration F_2 (kg/min)	restoration dep/acc (kg)
7	-70	10	15	10	+50
7	-70	10	18	15	+30
7	**-70**	10	18	10	**+80**
9	-90	10	15	10	+50
9	-90	10	18	15	+30
9	**-90**	10	18	10	**+80**

Table 3.4: Switching from result group 2 to 3 (example of upstream failure).

It is clear that for the 7 minute failure, the maximum amount of material that can be recovered during the restoration period (80 kg) is larger than the amount lost during the failure (70 kg). However, for the 9 minute failure, still only 80 kg can be recovered even though 90 kg have been depleted during the failure. Clearly this is not enough, and hence the optimizer must force an additional downstream shutdown (from any initial buffer level) to prevent more than 80 kg from ever being depleted. This 80 kg restoration limit can be calculated for the general case by using the difference between the maximum flow rate and minimum flow rate, as well as by using the restoration time, as given in the equation below,

$$\text{maximum material restored} = (F_{u,max} - F_{u,min})t_R. \tag{3.19}$$

3.3.2 3-Unit 2-Buffer System with Purge Option

Tables 3.5 and 3.6 outline the selected case study of a 3-unit 2-buffer system. Clearly, this case study is more complex than the 2-unit 1-buffer system presented earlier. The purpose of this case study is to highlight the importance of an optimization tool. In the previous study, only a small number of parameters were examined within a simple process system. In this simple study, the results were easy to understand and quantify. In the following study, however, a large number of parameters are studied simultaneously and the process size has increased. This increased complexity leads to complex results, which accordingly, are more difficult to quantify through simple linear equations.

Parameter	Units	Unit 1	Unit 2	Unit 3
$F_{u,min}$	kg/min	5	10	5
$F_{u,max}$	kg/min	18	15	20
$F_{u,0}$	kg/min	15	15	15
f_u	fraction	0.2	0.5	0.3
Failure duration $(t_2 - t_1)$	minutes	15	7	5
Recovery $(t_3 - t_2)$	minutes	10	5	8
C_u	$	2000	1500	500
R_U	$/kg	-	-	1
CP_u	$/kg	10	15	-

Table 3.5: 3-unit 2-buffer case study process unit parameters.

Parameter	Units	Buffer 1	Buffer 2
$B_{u,min}$	kg	0	0
$B_{u,max}$	kg	100	100

Table 3.6: 3-unit 2-buffer case study buffer parameters.

The parameters in Table 3.5 which include minimum, maximum, and initial flow rates are self-explanatory. Failure frequency, f_u, refers to the probability of failure. The sum of these fractions must add up to 1.0. Failure duration and recovery refer to the length of time for which a unit has a flow rate of 0 kg/min, and then the amount of time available for all buffers and flow rates to return to pre-shutdown values after the failed unit has been repaired. The fixed shutdown cost, C_u is the cost or penalty associated with any failure, regardless of the failure duration. The purge cost, CP_u, is a variable cost that is dependent on the amount of material purged from the system.

As the inclusion of this purge cost term indicates, this case study allows for material to be purged from process unit 1 or 2 without ever entering buffer 1 or 2 respectively. The purpose of allowing such purges is explained by an example. Suppose that process unit 2 has failed while units 1 and 3 continue operation. The unit 2 failure duration is such that buffer 1, which separates units 1 and 2, has filled completely. Process unit 1 now has the option to either shutdown temporarily or purge material temporarily until unit 2 is back online and buffer 1 is no longer at its maximum capacity. The optimization objective function is modified, as shown below, to include a cost associated with the amount of material that is purged from each unit in each scenario. Note that $FP_{u,s}(t)$ is the purge flow rate of unit u in scenario s at time t.

$$\max_{F_{U,s},B_{u,0}} \sum_{s=1}^{S} f_s\Big(\sum_{t=0}^{T} R_U F_{U,s}(t)\Delta t - \sum_{u=1}^{U} N_{u,s}C_u - \sum_{t=0}^{T}\sum_{u=1}^{U} CP_u FP_{u,s}(t)\Delta t\Big) \tag{3.20}$$

The results of this case study are represented by Figure 3.3. The optimal region is somewhat centrally located, and is bounded by the following five coordinates, with each representing the levels in buffer 1 and buffer 2 respectively: (30, 50), (65, 50), (65, 35), (40, 35), (30, 45). Due to the numerous parameters incorporated into this case study (see Tables 3.5 and 3.6) it is difficult to derive simple rules to quantify the observed optimal region as was done for the 2-unit 1-buffer case study.

The following points intuitively examine some of the key unit parameters individually.

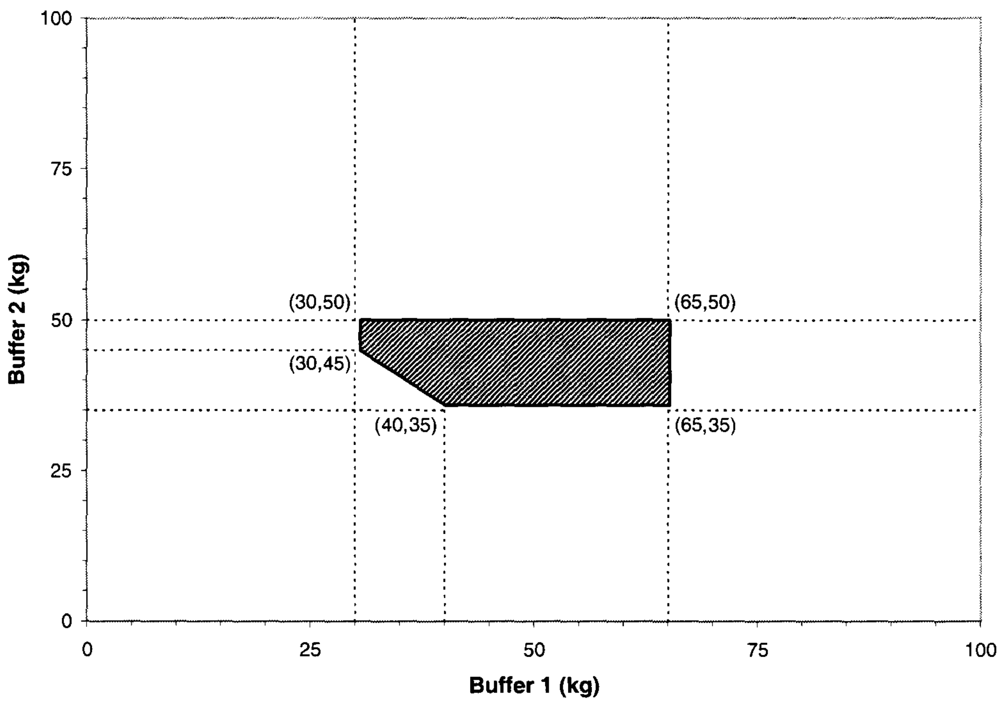

Figure 3.3: Optimal buffer region for the 3-unit 2-buffer case study.

- Comparison of the failure frequencies suggest that buffer 1 should be kept low (since unit 1 is less likely to fail than unit 2) while buffer 2 should be kept high (since unit 2 is more likely to fail than unit 3).

- According to failure duration, the level of buffer 1 is irrelevant (since unit 1 failure duration is so long that additional downstream shutdowns cannot be avoided from any initial level); however, the level of buffer 2 should be kept somewhat high (since unit 2 fails slightly longer than unit 3).

- Since unit 1 is more expensive to shutdown than unit 2, the level of buffer 1 should be kept low. Similarly, since unit 2 is more expensive to shutdown than unit 3, buffer 2 should be kept low.

Clearly, there are competing effects within only these three parameters mentioned. The remaining six unit parameters are more difficult to analyze intuitively; however, they may all have an impact on the optimal buffer levels. This still somewhat simple case study demonstrates that an optimization tool is required to handle growing problem complexity.

3.3.3 3-Tier Optimization

In order to uncover the 3-unit 2-buffer optimal region presented in Figure 3.3, a 3-tier optimization approach was required. In the first tier, the objective was to maximize plant economics, as defined by equation 3.1. The second tier involved extremizing one of the two initial buffer levels, while still matching the objective value obtained in tier 1. Finally, tier 3 extremized the remaining initial buffer level at both the maximum and minimum of the other buffer level, while again still matching the tier 1 objective value. To clarify this process, the 3-tier optimizations required to compute the five coordinates of Figure 3.3 are given in Table 3.7.

Tier 1	Tier 2	Tier 3	Resulting Coordinates
Maximize plant economics	minimize $B_{1,0}$	minimize $B_{2,0}$	(30,45)
Maximize plant economics	minimize $B_{1,0}$	maximize $B_{2,0}$	(30,50)
Maximize plant economics	maximize $B_{1,0}$	minimize $B_{2,0}$	(65,35)
Maximize plant economics	maximize $B_{1,0}$	maximize $B_{2,0}$	(65,50)
Maximize plant economics	minimize $B_{2,0}$	minimize $B_{1,0}$	(40,35)
Maximize plant economics	minimize $B_{2,0}$	maximize $B_{1,0}$	(65,35)
Maximize plant economics	maximize $B_{2,0}$	minimize $B_{1,0}$	(30,50)
Maximize plant economics	maximize $B_{2,0}$	maximize $B_{1,0}$	(65,50)

Table 3.7: 3-unit 2-buffer case study 3-tier optimization.

In addition to this 3-tiered approach, a grid search approach is also recommended. Completion of a grid search serves to increase confidence in results obtained by the multi-tier approach. In this case study, 121 coordinates on Figure 3.3 were manually tested to confirm the multi-tiered approach results.

3.3.4 Failure Duration Distribution

The next case study examined is that of a 2-unit 1-buffer system in which each of the two processing units have multiple failure modes. Recall that for the previous 2-unit case study, the upstream unit had one failure mode, as did the downstream unit. Here, each unit has five possible failure modes, and each failure mode has an associated frequency as well as failure duration (as depicted by Figure 3.4.)

As evident, the upstream unit may fail for 6, 8, 10, 12, or 14 minutes while the downstream unit may fail for one of 2, 4, 6, 8, or 10 minutes. These different failure durations could represent different failure mechanisms; for example, the 6 minute upstream failure could be a result of a mechanical disruption, while the 10 minute failure is due to an electrical failure, and the 14 minute failure is associated with an off-specification product. Alternatively, this distribution could simply represent uncertainty in the downtime of any one failure mechanism. It is important to examine the effect of such failure duration distributions to reflect realistic process shutdown characteristics. The failure modes selected to use in the optimization problem should be based on the most prominent failures according to a set of historical records for a certain process.

Table 3.8 conveys the same information as Figure 3.4. Although only five failure modes have been selected for each process unit, many more could be included to create any desired distribution. Tables 3.9 and 3.10 contain the remaining pertinent case study information. As indicated by Table 3.9, the option to purge material is

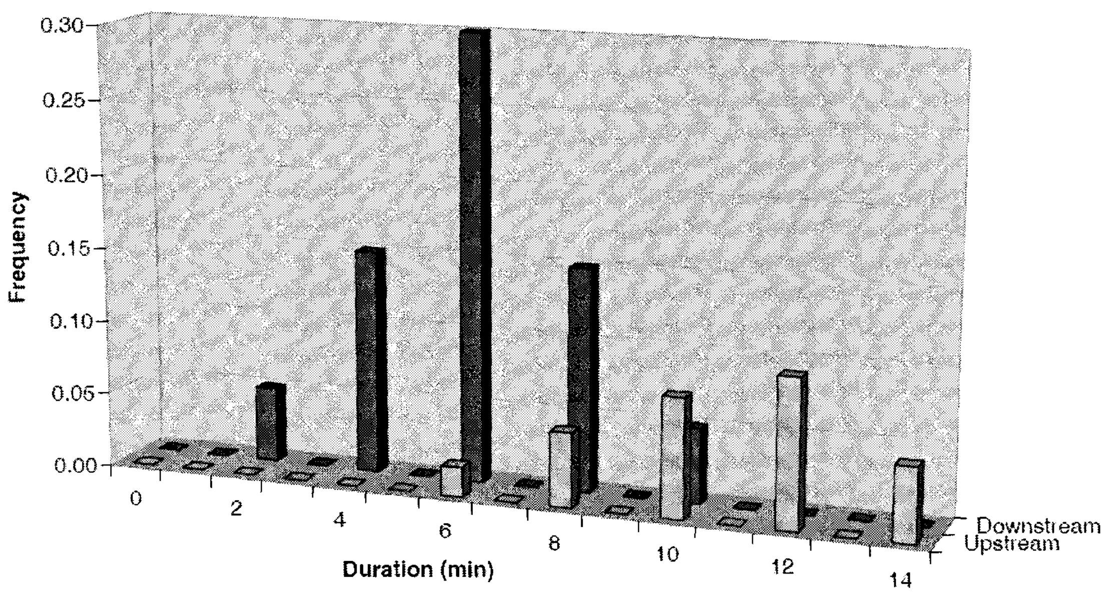

Figure 3.4: 2-unit 1-buffer case study with multiple failure modes for each process unit.

again included in this case study.

The result of this optimization was a unique solution, with 100 kg being the optimal nominal steady-state buffer level. This result, along with the maximum possible objective value obtained at various other nominal steady-state buffer levels are included as Figure 3.5. This unique solution is an interesting result, as both previous case studies revealed largely non-unique optima. Additionally, it is interesting to display these optimization results together as is done in Figure 3.5 to allow for human discretion when selecting the steady-state buffer level for implementation. An arbitrary horizontal line, simply for demonstrational purposes, has been drawn on Figure 3.5 to separate acceptable and unacceptable operating regions. In reality, this line should be drawn based on process-specific operational targets. Although the absolute optimum is at 100 kg in this case study, an examination of Figure 3.5 may result in a human decision to rather operate the process with the buffer level between 0 kg and perhaps

Scenario	Location	Failure Duration	Recovery	Failure frequency
		$t_2 - t_1$	$t_3 - t_2$	$f_{u,s}$
	(failed unit)	(min)	(min)	(fraction)
1	1 (upstream)	6	15	0.02
2	1 (upstream)	8	15	0.05
3	1 (upstream)	10	15	0.08
4	1 (upstream)	12	15	0.10
5	1 (upstream)	14	15	0.05
6	2 (downstream)	2	15	0.05
7	2 (downstream)	4	15	0.15
8	2 (downstream)	6	15	0.3
9	2 (downstream)	8	15	0.15
10	2 (downstream)	10	15	0.05

Table 3.8: Multi-failure mode shutdown scenario parameters.

Parameter	Units	Unit 1	Unit 2
$F_{u,min}$	kg/min	10	10
$F_{u,max}$	kg/min	18	18
$F_{u,0}$	kg/min	15	15
$\sum f_{u,s}$	fraction	0.3	0.7
C_u	$	2000	2000
R_U	$/kg	-	1
CP_u	$/kg	5	-

Table 3.9: Multi-failure mode case study process unit parameters.

30 kg. Since the objective value at 95 kg is very low, the risk of operating at 100 kg may not be acceptable. This decision of course would depend upon how tightly the buffer contents can be controlled. Also, selecting to operate at a lower buffer level would reduce inventory costs, which is often a major concern in industry.

Parameter	Units	Buffer 1
Minimum	kg	0
Maximum	kg	100

Table 3.10: Multi-failure mode case study buffer parameters.

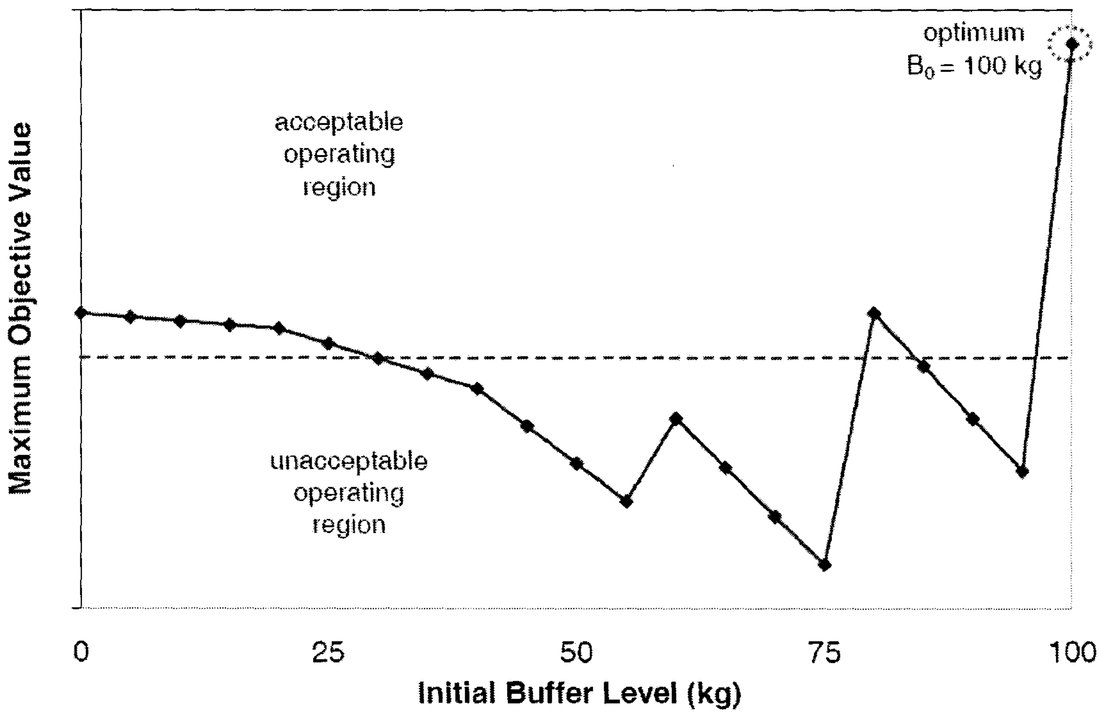

Figure 3.5: Optimization results for 2-unit 1-buffer case study with multiple failure modes.

The change in results from non-unique to unique solutions is related to the fact that each scenario is weighted into the objective function. In the past, with only one shutdown scenario for each unit, the upstream failure scenario would create a minimum buffer level while the downstream failure created a maximum buffer level. Here, five shutdown scenarios are possible for each the upstream and the downstream unit; hence, five upper buffer level limits and five lower limits are defined. Since each scenario is weighted into the objective function, operating at the lowest of the upper buffer levels is suboptimal compared to operating at the highest of the upper buffer levels. A further look at this case study clarifies this point.

The ten failure scenarios and their associated buffer limits are given in Table 3.11. These limits are determined from the maximum accumulation or depletion that can occur during the failure, while still avoiding an induced shutdown. Scenarios 4 and 5 are listed as "no effect" since more than 100 kg (i.e. the entire buffer contents) will be accumulated or depleted during these scenarios, and an additional shutdown cannot be avoided.

Scenario	Location (failed unit)	Failure Duration (min)	Associated Buffer Limit (kg)
1	upstream (1)	6	≥ 60
2	upstream (1)	8	≥ 80
3	upstream (1)	10	≥ 100
4	upstream (1)	12	no effect
5	upstream (1)	14	no effect
6	downstream (2)	2	≤ 80
7	downstream (2)	4	≤ 60
8	downstream (2)	6	≤ 40
9	downstream (2)	8	≤ 20
10	downstream (2)	10	≤ 0

Table 3.11: 2-unit 1-buffer multi-failure mode and associated buffer limits.

When examining the limits imposed by the upstream shutdowns, it is clear that satisfaction of the scenario 3 limit also involves satisfaction of scenarios 1 and 2. This is more optimal than operating at scenario 1 or 2, where at least 1 upper limit is not fulfilled. Accordingly, three "spikes" of increasing magnitude are formed on Figure 3.5: at 60 kg where scenario 1 satisfied, at 80 kg where scenarios 1 and 2 are satisfied, and at 100 kg where scenarios 1 through 3 are satisfied.

Similarly, the lower limit imposed by scenario 10 is sufficient to fulfill the limits imposed by scenarios 6 through 9. It would clearly be sub-optimal to operate in any of the limits imposed by scenarios 6 through 9 limits, as at least the constraint imposed by scenario 10 would not be satisfied. Accordingly, the objective value trend line in Figure 3.5 is decreasing as the initial buffer level increases from 0 kg, where scenarios 6 through 10 are satisfied, to 20 kg and 40 kg, where less scenarios are satisfied. This trend changes upon reaching 60 kg, since both upstream and downstream failure scenarios are affected by nominal levels between 60 kg and 80 kg.

A second failure distribution case study is briefly examined to demonstrate the various optimal solution configurations. Again, each unit has five possible failure modes, and each failure mode has an associated frequency as well as failure duration. In this case study, product purge is not an option.

Table 3.12 conveys the failure distribution information. Although only five failure modes have been selected for each process unit, many more could be included to create any desired distribution. Tables 3.13 and 3.14 contain the remaining pertinent case study information.

The result of this optimization was a unique solution, with 70 kg being the optimal steady-state buffer level. This result, along with the maximum possible objective value obtained at various other buffer levels are included as Figure 3.6. This solution shows a clearly different trend than that of Figure 3.5, as the optimum is not located at an extreme of the buffer limits. Here again, an arbitrary line has been drawn that separates acceptable and unacceptable operating regions. These two case studies clearly reveal the increased solution complexity and the need for an optimization tool, since it would be insufficient to rely on simple linear relationships as presented earlier for the simple case study.

Scenario	Location	Failure Duration	Recovery	Failure frequency
		$t_2 - t_1$	$t_3 - t_2$	$f_{u,s}$
	(failed unit)	(min)	(min)	(fraction)
1	1 (upstream)	3	15	0.05
2	1 (upstream)	4	15	0.05
3	1 (upstream)	5	15	0.10
4	1 (upstream)	6	15	0.10
5	1 (upstream)	7	15	0.20
6	2 (downstream)	1	15	0.20
7	2 (downstream)	2	15	0.10
8	2 (downstream)	3	15	0.10
9	2 (downstream)	4	15	0.05
10	2 (downstream)	5	15	0.05

Table 3.12: 2-unit 1-buffer multi-failure mode shutdown scenario parameters.

Parameter	Units	Unit 1	Unit 2
$F_{u,min}$	kg/min	10	10
$F_{u,max}$	kg/min	18	18
$F_{u,0}$	kg/min	15	15
$\sum f_{u,s}$	fraction	0.5	0.5
C_u	$	2000	2000
R_U	$/kg	-	1

Table 3.13: 2-unit 1-buffer multi-failure mode case study process unit parameters.

Parameter	Units	Buffer 1
$B_{u,min}$	kg	0
$B_{u,max}$	kg	100

Table 3.14: 3-unit 2-buffer case study buffer parameters.

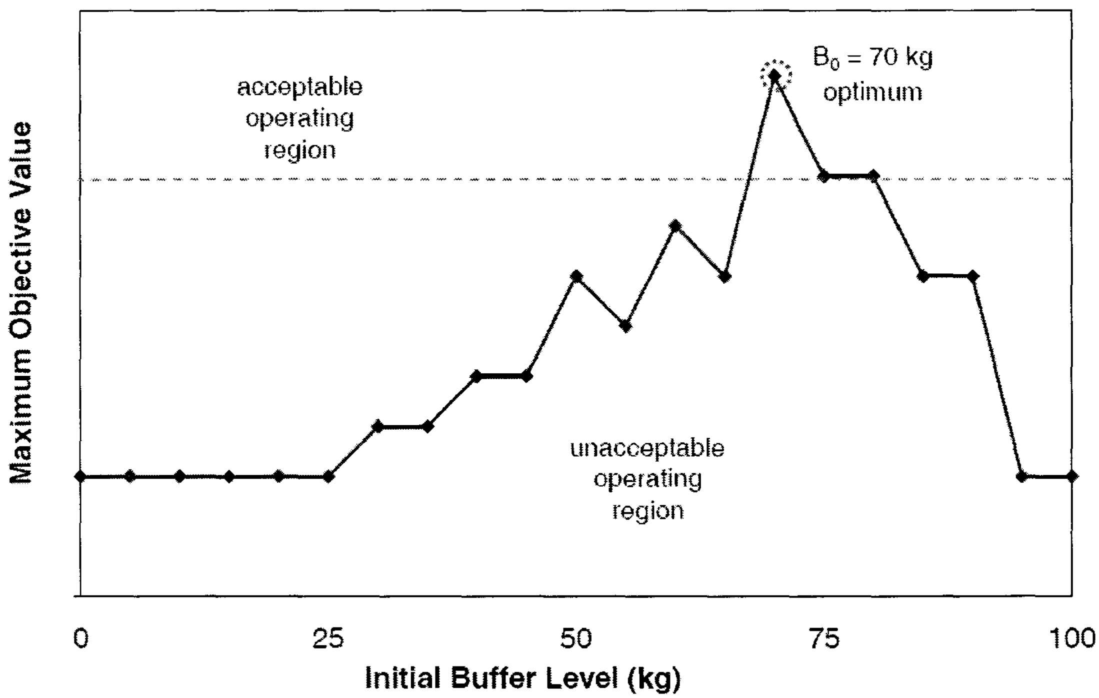

Figure 3.6: Optimization results for 2-unit 1-buffer case study with multiple failure modes.

3.3.5 Soft-Constraint Approach

An alternative to the mixed-integer formulation is now examined. A soft constraint approach used to penalize flow rates that violate the minimum bound for viable operation and the cost of a failure or shutdown is accounted for in terms of the loss of production and a penalty - no fixed shutdown cost is incorporated.

The objective function in this formulation is a modified version of equation 3.1. Profit is still associated with the summation of the furthest downstream flow rate; however, the fixed shutdown cost is replaced by the minimum flow rate constraint violation, ϵ, multiplied by a constant, C, for relative weighting,

$$\max_{F_{U,s},B_{u,0}} \sum_{s=1}^{S} f_s\Big(\sum_{t=0}^{T} R_U F_{U,s}[t]\Delta t - \sum_{t=0}^{T}\sum_{u=1}^{U} C\epsilon_{u,s}[t]\Delta t\Big). \tag{3.21}$$

The optimization is subject to the following additional constraints,

$$F_{u,min} - \epsilon_{u,s}[t] \leq F_{u,s}[t] \tag{3.22}$$

$$F_{u,s}[t] \geq 0 \tag{3.23}$$

$$\epsilon_{u,s}[t] \geq 0. \tag{3.24}$$

Figure 3.7 illustrates the definition of the term ϵ. Clearly, the inclusion of ϵ in the objective function is done with the purpose of keeping the unit flow rates at or above the desired minimum.

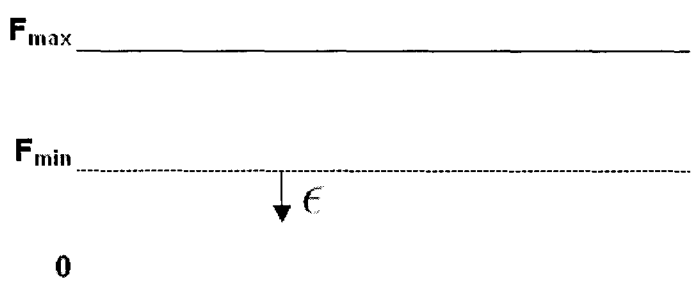

Figure 3.7: Depiction of the term epsilon, ϵ.

This formulation has both advantages and disadvantages in comparison to the mixed-integer approach. The removed requirement for integer variables is a computational advantage; integer variables significantly increase problem complexity and hence solution time. Unfortunately it is difficult to correctly account for the cost of a failure or shutdown with this method. Despite this drawback, it is possible that this method may give reasonably agreeable optimal solutions to those of the mixed-integer approach. A case study is examined to compare the results for these alternate formulation approaches.

Soft-Constraint Case Studies

Two case studies have been selected for comparison of the soft-constraint and mixed-integer formulations. It is clear from the problem formulations that it would be difficult or impossible to select parameter values that would make them entirely equivalent. However, process parameter values were kept the same.

The first case study is associated with the first case study of Section 3.3.4 (failure duration distribution). For comparison purposes, please refer to Figure 3.4 and Tables 3.8, 3.9 and 3.10. The soft-constraint case study uses all of the same parameters as listed in the Section 3.3.4 case study, with the exception of the fixed shutdown cost, C_u, parameter. In Table 3.9, these values were listed as 2000. In this case study, no such parameter exists. In this soft-constraint formulation a fixed weight, C, is included in the objective function to reflect the importance of keeping flow rates at or above the minimum flow rate. Here, this parameter is assigned a value of 10. Figure 3.8 presents the optimization results.

A quick comparison of Figure 3.8 with Figure 3.5 reveals substantially different results. A different overall trend, as well as a different optimum are found in the soft-constraint approach. Although additional results are not included here, a sensitivity test revealed the effect of the constant C term in the objective function. An increased value leads to a more concave trend line, where as a decreased value lead to a more linear and positive relationship between initial buffer level and maximum objective value.

The second case study pertains to the second case study of Section 3.4, as detailed by Tables 3.12, 3.13 and 3.14. Again, all of the same information applies to this

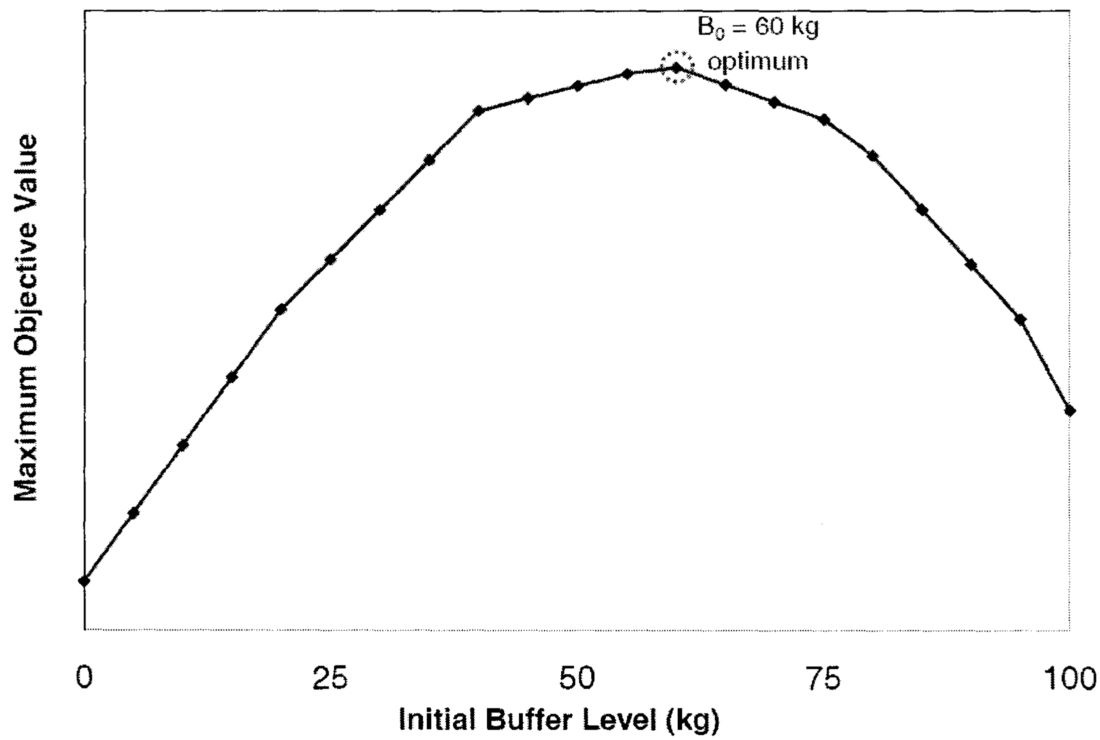

Figure 3.8: Optimization results for soft-constraint case study.

soft-constraint approach, except for the fixed shutdown cost, C_u, parameter. This parameter is not included in the soft-constraint formulation, but instead a fixed cost, $C = 10$, is incorporated to increase the importance of the flow rate constraints. The results of this soft-constraint case study are presented as Figure 3.9, and should be compared to the mixed-integer case study results of Figure 3.6.

Again, a concave trend line is observed. In this case, however, this shape is quite similar to that of the mixed-integer case study. In addition, the same optimum is found. A sensitivity analysis on the C parameter revealed a change in objective value magnitudes, however, the trend line shape remained the same, as did the optimum.

The two soft-constraint case studies examined revealed noticeably different results. The first case study did a poor job of "approximating" the associated mixed-integer

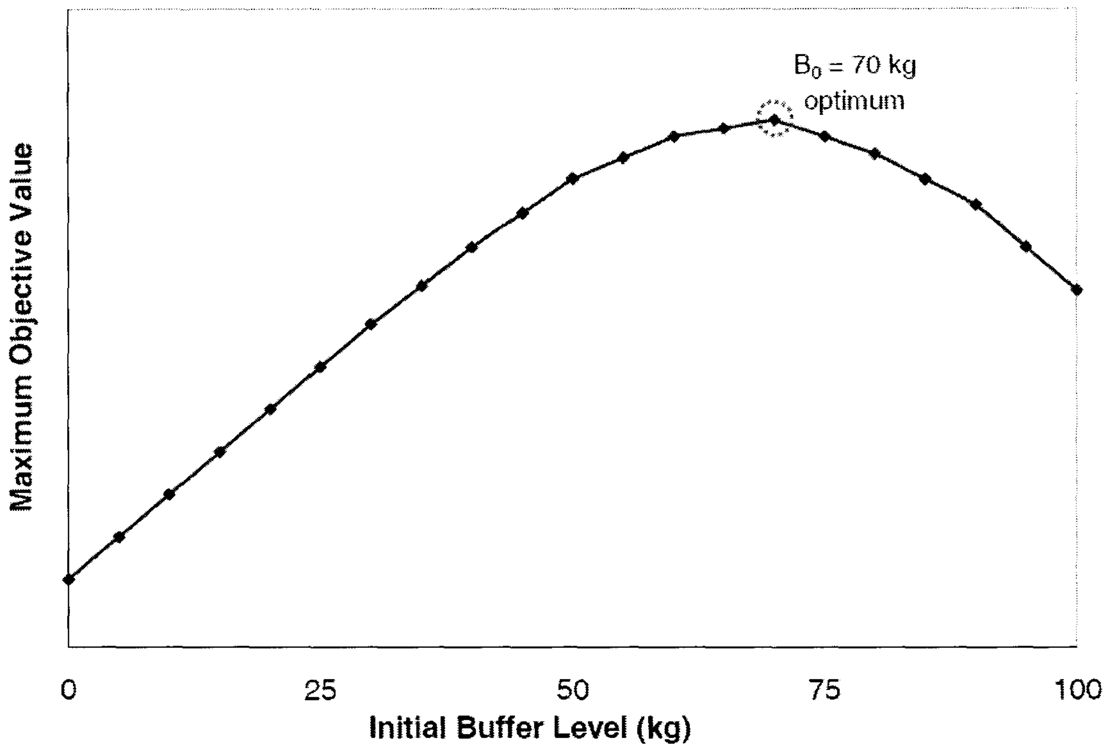

Figure 3.9: Optimization results for soft-constraint case study.

results; both the trend line and optimum were incorrect, while the second case study resulted in an appropriate trend line shape and the correct optimum. While removing the fixed shutdown cost constraint results in a significantly simpler optimization problem with more smooth optimization results, further investigation is needed to evaluate the accuracy of the approximation and the conditions under which it may be safely applied.

3.3.6 Pulp Mill Case Study

A pulp mill consisting of seven processing units, six intermediate storage tanks, and a recycle stream is now considered. The model, based on the information presented by Leiviska *et al.* (1980), is illustrated by Figure 3.10.

The steam balances and modelling completed in Leiviska *et al.* (1980) are not included here. Notation and information on the process stream numbers, process units, buffer tanks and buffer sizes are presented in Figure 3.10. Sufficient information was available in Leiviska *et al.* (1980) to extract approximate flow rates and material consumption or generation at each processing stage. The ratio of fresh pulp feed to liquor recycle has been arbitrarily selected.

This system was modelled via the mixed-integer approach. All formulation methodology presented earlier applies to this case study, but with one exception. Since information on approximate material consumption and generation information was available, a linear relationship exists between the inlet and outlet flow rates of processing units as follows:

$$F_2(t) = F_3(t)b_6 \tag{3.25}$$

$$F_7(t) = F_3(t)b_2 \tag{3.26}$$

$$F_5(t) = F_4(t)b_1 \tag{3.27}$$

$$F_9(t) = F_8(t)b_3 \tag{3.28}$$

$$F_{11}(t) = F_{10}(t)b_4 \tag{3.29}$$

$$F_{13}(t) = F_{12}(t)b_5 \tag{3.30}$$

where $b_1 = 1.25, b_2 = 1.25, b_3 = 0.25, b_4 = 2.00, b_5 = 1.50$ and $b_6 = 2.25$. These relationships are easily incorporated into the mass balance.

To minimize computational complexity, only the digester and bleaching units are considered unreliable (and therefore prone to unplanned failure) in this case study. The failure distribution is given in Table 3.15. Note that the relative shutdown cost associated with all six of the processing units was given a value of 1.0. Overall, the digester and bleaching units are equally likely to fail.

Table 3.16 presents the process stream information. The nominal steady-state flow

Scenario	Location (failed unit)	Failure Duration $t_2 - t_1$ (min)	Recovery $t_3 - t_2$ (min)	Failure frequency $f_{u,s}$ (fraction)
1	Digester	5	5	0.05
2	Digester	10	5	0.05
3	Digester	20	5	0.10
4	Digester	30	5	0.10
5	Digester	40	5	0.20
6	Bleaching	5	5	0.20
7	Bleaching	10	5	0.10
8	Bleaching	20	5	0.10
9	Bleaching	30	5	0.05
10	Bleaching	40	5	0.05

Table 3.15: Pulp mill shutdown scenario parameters.

rates were approximated from production schedule information provided by Leiviska *et al.* (1980), and all minimum and maximum flow rates were selected as 35% and 125% of the initial flow rate.

Process Stream	$F_{u,0}$ (m^3/hr)	$F_{u,max}$ (m^3/hr)	$F_{u,min}$ (m^3/hr)
1	105	131.25	36.75
2	180	225	63
3	80	100	28
4	80	100	28
5	100	125	35
6	100	125	35
7	100	125	35
8	100	125	35
9	25	31.25	8.75
10	25	31.25	8.75
11	50	62.5	17.5
12	50	62.5	17.5
13	75	93.75	26.25
14	75	93.75	26.25

Table 3.16: Pulp mill process stream parameters.

The optimal nominal steady-state buffer levels are given in Table 3.17 alongside their respective limits. Note that the results were obtained with a computation time of slightly less than 1 minute.

Buffer u	Capacity (m^3)	$B_{u,opt}$ (m^3)	Optimal / Capacity (%)
1	1500	625	42
2	1500	360	24
3	2000	500	25
4	800	156.25	20
5	1600	1475	92
6	2000	1531.25	77

Table 3.17: Pulp mill optimal initial buffer levels.

Tier 2: Inventory Minimization

As was the case with several previous case studies, the results of the pulp mill case study were non-unique. A second tier optimization with an objective of inventory minimization was completed to demonstrate the non-uniqueness and to achieve a second optimization goal. In this second tier, the initial objective is maintained by enforcing:

$$\sum_{s=1}^{S} f_s\Big(\sum_{t=0}^{T} R_U F_{U,s}(t)\Delta t - \sum_{u=1}^{U} N_{u,s} C_u\Big) \geq Obj_{Tier1} \tag{3.31}$$

where Obj_{Tier1} is the maximum objective value obtained from the first tier optimization. The objective function of the second tier is given by:

$$\min \sum_{u=1}^{U-1} B_{opt,u}. \tag{3.32}$$

Note that if desired, importance weighting parameters could be used to reflect inventory minimization priorities.

Computational effort for the second tier was minimized by specifying the result of the binary variables from the first tier as parameters in the second tier ($w_{u,s}[t]$). The results of this second tier are compared alongside the initial results in Table 3.18. As demonstrated, this two-tiered approach allows for further plant economics minimization.

Buffer u	Capacity (m^3)	Tier 1 $B_{u,opt}$ (m^3)	Tier 2 $B_{u,opt}$ (m^3)
1	1500	625	624.8
2	1500	360	280
3	2000	500	499.8
4	800	156.25	156.25
5	1600	1475	192.5
6	2000	1531.25	288.75

Table 3.18: Pulp mill optimal initial buffer levels.

This case study demonstrated the ability of the mixed-integer linear programming formulation to handle larger and more complex problems. In addition, although the set of optimal initial levels obtained was non-unique, a second optimization tier was completed to achieve inventory minimization while still maintaining the tier 1 objective.

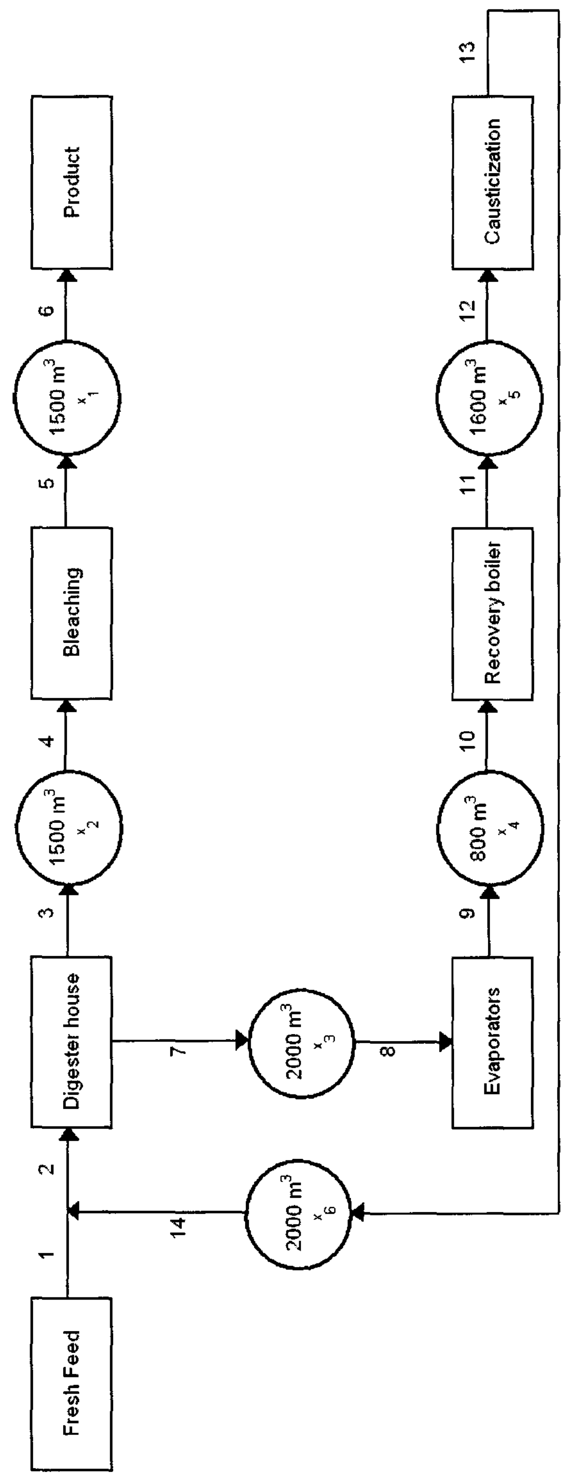

Figure 3.10: Leiviska *et al.* (1980) pulp mill case study.

3.4 Chapter Summary

In any given manufacturing process, unit shutdowns - whether planned or unplanned - are inevitable. In the unplanned case, the availability of historical records (such as mean time between failure (MTBF) data or other failure probabilities) provides insight which can be used to minimize shutdown impacts on the overall process through the use of intermediate storage. Several case studies have been presented which apply to general process configuration of n units separated by $n-1$ buffer tanks. The goal of this work was to determine the economical optimal *nominal* buffer level, based on knowledge of various probable failure scenarios. These problems were posed as multi-scenario MILPs, coded in AMPL and solved by CPLEX.

A simple 2-unit 1-buffer case study was examined in depth, and results were explained and modelled by simple linear relationships. The increased complexity of the remaining test cases (3-unit 2-buffer with purge, 2-unit 1-buffer with failure distributions) prevented such simple explanations, but rather demonstrated the need for an advanced optimization to uncover the optimal nominal buffer levels.

Next, a soft-constraint approach was presented as an alternate formulation which did not require the use of integer variables. Comparison of two soft-constraint case studies with the original formulation had mixed results; one case closely resembled the mixed-integer results while the other did not. This discrepancy is attributed to the failure of the soft-constraint approach to accurately reflect the cost of a unit shutdown.

Finally, a pulp mill case study was presented. This case study demonstrated the ability of the mixed-integer linear programming formulation to handle larger and more complex problems.

Chapter 4

Flexible Maintenance Scheduling

This chapter introduces the topics of production and maintenance scheduling. Next, a mixed-integer linear programming formulation is presented to optimally schedule both production and maintenance, followed by a presentation and discussion of various case studies. The chapter concludes with a summary. All case studies in this chapter were implemented in AMPL, and solved by CPLEX.

4.1 Introduction

The goal of this chapter is to optimally and simultaneously schedule required maintenance activities and production over a given time horizon. The continuous and dynamic nature of the process considered dictates that intermediate storage levels must also be accounted for during product changeover or maintenance, hence dynamic material balances are included. In addition, this problem is complicated by the existence of sequence-dependent set-up times between products (a reflection of machine alterations or cleaning), and finite intermediate storage. Maintenance resources are not considered as a constraint in this problem due to the minimal maintenance requirements enforced.

We present first a formulation derived primarily for batch systems, as it involves relatively straightforward adaptation of the scheduling formulation of Kondili *et al.* (1993). It is directly applicable to the problem under consideration under conditions of no splitting of tasks; relaxation of this assumption is possible, but involves the introduction of more tasks with a corresponding increase in problem dimension. We therefore present next an alternative formulation that is more suited to continuous processes, but incorporates many of the formulation ideas and constructs of the original batch formulation.

4.2 Batch Process Formulation

This formulation is based on the general scheduling algorithm presented by Kondili *et al.* (1993). This algorithm relies on the concept of state-task-networks (STN), which was developed to remove ambiguities from process diagrams and introduced in Chapter 2. Although the process studied here is simple, and hence does not necessarily need a STN to describe it, the STN method is still followed in order to be consistent with the scheduling algorithm and to maintain a general formulation applicable to more complex process configurations. Figure 4.1 depicts the general 2-unit 1-buffer process in the STN framework. States are denoted by s, tasks by i, and processing units by j.

Figure 4.1 describes the path of one set of raw materials through the process. In this application, the aim is to schedule a multi-product plant, hence several STNs are required to differentiate between the various raw materials, variation in processing times, and intermediate and final products. Figure 4.2 depicts five "jobs" or the path of five feed stocks into final products. Note that each job has two tasks associated

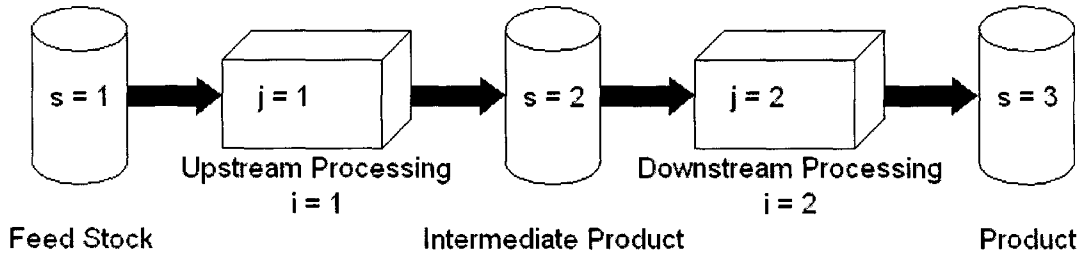

Figure 4.1: Depiction of State-Task-Network applied to the studied 2-unit 1-buffer process.

with it - one for the upstream processing task, and one for the downstream processing task.

4.2.1 Production Scheduling

The first variable introduced is an indicator variable, $W_{i,j}[t]$. This variable takes a value of one if at time t equipment j starts task i, or takes a value of zero otherwise. The following allocation constraint is used to ensure that at most, one task i may begin in unit j at time t. Note that I_j is the set of tasks that can be completed by unit j.

$$\sum_{i \epsilon I_j} W_{i,j}[t] \leq 1 \qquad \forall j,\, t \qquad W_{i,j}[t] \,\epsilon\, \{0,1\} \tag{4.1}$$

To ensure that unit j does not begin another task until the current task is finished, the additional allocation constraint is enforced,

$$\sum_{i' \epsilon I_j} \sum_{t'=t}^{t+p_i-1} W_{i',j}[t'] - 1 \leq M(1 - W_{i,j}[t]) \qquad \forall j,\, t,\, i \,\epsilon\, I_j \tag{4.2}$$

where M is a sufficiently large positive number, and p_i is the processing time for task

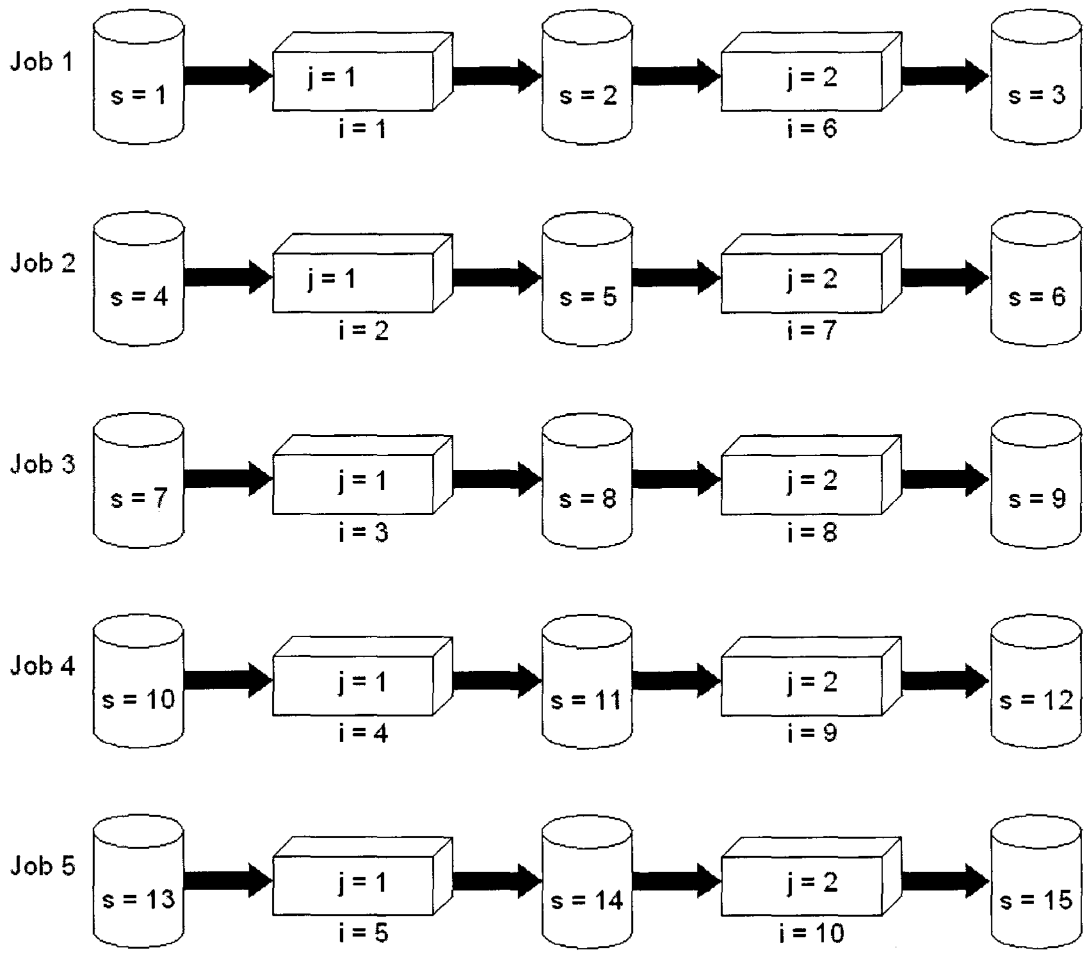

Figure 4.2: Depiction of five State-Task-Networks for the 2-unit 1-buffer process.

i.

In this problem it is assumed that feed stock and product storage are unconstrained. Accordingly, the capacity limitations for the feed stock states and the product states are bounded by

$$0 \leq S_s[t] \leq M \qquad \forall\, s \,\epsilon\, D_f,\; s \,\epsilon\, D_p,\; t \tag{4.3}$$

where as before, M is a large, positive number. $S_s[t]$ is the amount of material in state s at time t, and D_f and D_p are the sets of feed states and product states respectively.

The intermediate products ($s \,\epsilon\, D_i$) must conform to a more rigid set of capacity constraints to reflect the fact that there is only one shared storage buffer tank, and that this storage space is finite.

$$0 \leq S_s[t] \leq V_s[t] C_s \qquad \forall\, s \,\epsilon\, D_i,\; t \tag{4.4}$$

$$\sum_{s \epsilon D_i} V_s[t] \leq 1 \qquad V_s[t] \,\epsilon\, \{0, 1\} \qquad \forall\, t \tag{4.5}$$

Equation 4.4 ensures that the storage capacity is not exceeded, while equation 4.5 ensures that only one intermediate product (state) is in the buffer at a time. $V_s[t]$ is a binary indicator variable that takes a value of one when state s is stored in the intermediate buffer tank at time t and takes a value of zero otherwise. C_s is the maximum amount of state s that can be stored in the intermediate storage tank. This shared storage formulation is based on the work by Maravelias and Grossmann (2003).

For any batch process, a material balance to compute the amount of material in any state (feed stock, intermediate, or product state) can be written as

$$S_s[t] = S_s[t-1] + \sum_{i \epsilon \bar{T}_s} \bar{\rho}_{i,s} \sum_{j \epsilon K_i} B_{i,j}[t - p_i] - \sum_{i \epsilon T_s} \rho_{i,s} \sum_{j \epsilon K_i} B_{i,j}[t] \qquad (4.6)$$

$$\forall\, s,\, t$$

where $\bar{\rho}_{i,s}$ is the fraction of material, in state s, produced from task i, and $\rho_{i,s}$ is the fraction of material, in state s, required as input to task i. $B_{i,j}[t]$ represents the amount of material of task i that begins processing in unit j at time t. K_i is the set of units that can perform task i, $\bar{T}_s$ is the set of tasks producing material in state s, and T_s is the set of tasks receiving material from state s. The net increase in the amount of material in state s at time t is equal to the difference of the amount of material produced in s and the amount used in state s.

Kondili *et al.* (1993) have provided an extension which more closely models continuous processes by introducing the parameters $\rho_{i,s}[\theta]$, $\bar{\rho}_{i,s}[\theta]$, and λ_i. The term $\rho_{i,s}[\theta]$ refers to the fraction of material, in state s, required as input to task i at the beginning of time interval θ relative to the start of the task. Similarly, $\bar{\rho}_{i,s}[\theta]$ is the fraction of material, in state s, produced from task i at the beginning of time interval θ relative to the start of the task. So although one must specify a "batch" size, the batch can be "continuously" fed into and withdrawn from the processing unit over the processing time, p_i, in intervals from $\theta = 0, ..., p_i$. The mass balance, with this modification becomes:

$$S_s[t] = S_s[t-1] + \sum_{i \epsilon \bar{T}_s} \sum_{j \epsilon K_i} \sum_{\theta=0}^{p_i} \bar{\rho}_{i,s}[\theta] B_{i,j}[t-\theta] - \sum_{i \epsilon T_s} \sum_{j \epsilon K_i} \sum_{\theta=0}^{p_i} \rho_{i,s}[\theta] B_{i,j}[t-\theta] \qquad (4.7)$$

$$\forall\, s,\, t.$$

It is important to note the normalization requirement of $\rho_{i,s}[\theta]$ and $\bar{\rho}_{i,s}[\theta]$, as given by:

$$\sum_{s \epsilon S_i} \sum_{\theta=0}^{p_i} \rho_{i,s}[\theta] = \sum_{s \epsilon \bar{S}_i} \sum_{\theta=0}^{p_i} \bar{\rho}_{i,s}[\theta] = 1 \qquad \forall\, i. \tag{4.8}$$

S_i refers to the set of input states of task i and $\bar{S}_i$ refers to the set of output states of task i. In the case study presented here, each task has only one input state and one output state.

Example

Suppose that 100 kg of material begin processing (as task 1) in equipment unit 1 at the beginning of time period 1. The intermediate state 2 then begins processing (as task 6) in equipment unit 2 at time period 7. Referring to Figure 4.2, Table 4.1 computes the amount of material in state 2 according to how much is produced by task 1 and how much is consumed by task 6 (given by equation 4.7). As mentioned, in this example each task has only one input state and one output state; hence equation 4.7 for this example becomes the following for state 2:

$$S_2[t] = S_2[t-1] + \sum_{\theta=0}^{p_1} \bar{\rho}_{1,2}[\theta] B_{1,1}[t-\theta] - \sum_{\theta=0}^{p_6} \rho_{6,2}[\theta] B_{6,2}[t-\theta] \qquad \forall\, t$$

Due to the batch nature of the Kondili *et al.* (1993) formulation, capacity limitations must be set for each task. $V_{i,j}^{min}$ and $V_{i,j}^{max}$ refer to the minimum and maximum amount of material that can be contained in equipment unit j for task i, giving the constraint,

$$W_{i,j}[t] V_{i,j}^{min} \leq \lambda_i B_{i,j}[t] \leq W_{i,j}[t] V_{i,j}^{max} \qquad \forall\, i,\, j,\, t \tag{4.9}$$

t	θ_1	$\bar{\rho}_{1,2}[\theta_1]$	$B_{1,1}[t]$	θ_6	$\rho_{6,2}[\theta_6]$	$B_{6,2}[t]$	$S_2[t]$
1	0	0	100	-	-	-	0
2	1	0.125	0	-	-	-	12.5
3	2	0.125	0	-	-	-	25
4	3	0.125	0	-	-	-	37.5
5	4	0.125	0	-	-	-	50
6	5	0.125	0	-	-	-	62.5
7	6	0.125	0	0	0.2	100	55
8	7	0.125	0	1	0.2	0	47.5
9	8	0.125	0	2	0.2	0	40
10	-	-	0	3	0.2	0	20
11	-	-	0	4	0.2	0	0
12	-	-	0	5	0	0	0
13	-	-	0	-	-	0	0
14	-	-	0	-	-	0	0

Table 4.1: Batch material balance example for state 2.

where λ_i is the maximum fraction of "batch" i that ever resides in a processing unit. This parameter may be calculated from the expression (Kondili et al, 1993),

$$\lambda_i = \max_{0 \leq \theta \leq p_i} \sum_{\theta'=0}^{\theta} \Big(\sum_{s \,\epsilon\, S_i} \rho_{i,s}[\theta'] - \sum_{s \,\epsilon\, \bar{S}_i} \bar{\rho}_{i,s}[\theta'] \Big) \qquad \forall i. \tag{4.10}$$

The specification of $\rho_{i,s}[\theta]$ and $\bar{\rho}_{i,s}[\theta]$ and the subsequent calculation of λ_i can and should be completed before the equipment unit capacity limitations are enforced.

To account for customer demands (or product completion deadlines), the following constraint is included:

$$S_s[t_{demand,s}] \geq D_s[t_{demand,s}] \qquad \forall\, s \,\epsilon\, D_p \tag{4.11}$$

where $D_s[t_{demand,s}]$ is the amount of state s required (demanded) at time interval $t_{demand,s}$. This constraint requires the amount of material in the final product state at the demand time, $t_{demand,s}$, to satisfy specified customer or production requirements. Alternatively, this product completion constraint could be implemented as a soft constraint. This soft-constraint approach is examined as a case study in Section 4.3.3.

Sequence-dependent cleaning is incorporated to ensure that if a task within family k begins processing, the following task cannot start until at least after $\tau_{j,k,k'}$ time intervals, where the latter task is in family k'. This is written mathematically as

$$\sum_{i' \epsilon I_j^{(k')}} \sum_{t'=t+p_i}^{t+p_i+\tau_{j,k,k'}-1} W_{i',j}[t'] \leq M\Big(1 - W_{i,j}[t]\Big) \qquad \forall j,\ i \epsilon I_j^{(k)},\ t \tag{4.12}$$

$$\sum_{i' \epsilon I_j^{(k \backslash i)}} \sum_{t'=t+p_i}^{t+p_i+\tau_{j,k,k}-1} W_{i',j}[t'] \leq M\Big(1 - W_{i,j}[t]\Big) \qquad \forall j,\ i \epsilon I_j^{(k)},\ t \tag{4.13}$$

where $I_j^{(k')}$ is the set of tasks that belong to family k' and can be completed by equipment unit j, and where $I_j^{(k)}$ is the set of tasks that belong to family k and can be completed by equipment unit j. The term $\tau_{j,k,k'}$ is the amount of time required to clean equipment unit j when switching from a task of family k to a task of family k'. Equation 4.13 and is also required to enforce cleaning between products of the same family. Clearly in practice $\tau_{j,k,k} \leq \tau_{j,k,k'}$. Note that $k \backslash i$ is the set of all tasks belonging to family k except the current task i.

Lastly, the continuous variable T_{ms} is introduced in order to identify the makespan (completion time of the last scheduled job). This final completion time is calculated by

$$T_{ms} \geq W_{i,j}[t](t + p_i - 1) \qquad \forall i,\ j,\ t \tag{4.14}$$

which is typically minimized in the objective function:

$$\min_{W_{i,j}[t]} T_{ms}. \tag{4.15}$$

An additional constraint is enforced to ensure that the entire batch is processed together; so that each task can only be started once (i.e. one job cannot be split into sub-jobs):

$$\sum_{i \epsilon T_s} \sum_{j \epsilon K_i} \sum_{t=1}^{H} W_{i,j}[t] \leq 1 \qquad \forall\, s \tag{4.16}$$

This constraint restricts states to be processed by only one task and one unit, and ensures a constant"processing rate". Without inclusion of equation 4.16, one batch could be split into many smaller batches, but the processing time would remain constant regardless of batch size.

4.2.2 Extension to Maintenance Scheduling

The objective in regards to maintenance scheduling is to have one or more maintenance tasks scheduled between production jobs with a specified time slot within which maintenance must begin. The maintenance constraints must be constructed carefully to allow flexibility when assigning the starting time of maintenance events (within the specified time slot), and also to ensure that maintenance will begin within this time slot.

The following constraint ensures that maintenance cannot occur while equipment unit j is processing task i,

$$\sum_{t'=t}^{t+p_i-1} M_j[t'] \leq M(1 - W_{i,j}[t]) \qquad \forall\, j,\; i \,\epsilon\, I_j,\; t = 1, \ldots, H - p_i + 1. \tag{4.17}$$

The term $M_j[t]$ is a binary indicator variable which is assigned a value of one if maintenance is occurring on unit j at time t, and a value of zero otherwise. Here again, M is a sufficiently large, positive number.

Next, to enforce that a second maintenance task may not begin until the first maintenance task is complete, the following is applied,

$$\sum_{t'=t}^{t+T_{j,m}-1} \sum_{m'=1}^{M_{tot,j}} x_{j,m'}[t'] - 1 \leq M(1 - x_{j,m}[t]) \qquad \forall j,\, m \qquad t = 1, \ldots, H - T_{j,m} + 1. \tag{4.18}$$

$x_{j,m}[t]$ is a binary indicator variable that is assigned a value of one if unit j begins maintenance event m at time period t, and is assigned a value of zero otherwise. $T_{j,m}$ is the required number of consecutive time intervals for maintenance event m in unit j, and $M_{tot,j}$ is the total number of required maintenance events for unit j. Although it is likely not desirable in reality, this formulation will allow two maintenance events to occur consecutively (if found to be optimal).

The following constraint determines the total number of maintenance operations that occur on unit j over a given time horizon,

$$\sum_{t=1}^{H} \sum_{m=1}^{M_{tot,j}} x_{j,m}[t] = M_{tot,j} \qquad \forall j \tag{4.19}$$

where $M_{tot,j}$ specifies the number of desired maintenance events. Alternatively, the equality may be replaced by an inequality, and hence the user can force a specific maximum or minimum number of maintenance events to occur.

A consecutive maintenance period of at least $T_{j,m}$ time intervals is enforced through

$$\sum_{t'=t}^{t+T_{j,m}-1} M_j[t'] \geq T_{j,m}\, x_{j,m}[t] \qquad \forall\, j,\, m,\, t = 1, \ldots, H - T_{j,m} + 1. \tag{4.20}$$

This constraint ensures that once a maintenance event begins, it cannot be interrupted.

By specifying a set of consecutive time intervals, T_{mk}, specific to each maintenance event, m, the following can be utilized to enforce that maintenance can and must begin within this interval.

$$\sum_{t \,\epsilon\, T_{mk}} x_{j,m}[t] \geq 1 \qquad \forall\, j,\, m \tag{4.21}$$

The objective function remains unchanged, as does the makespan calculation; however, they are repeated below for completeness.

$$T_{ms} \geq W_{i,j}[t](t + p_i - 1) \qquad \forall\, i,\, j,\, t \tag{4.22}$$

$$\min_{W_{i,j}[t]} T_{ms} \tag{4.23}$$

4.3 Batch Process Case Studies

Several case studies are now examined. First, a makespan minimization is conducted, followed by a throughput maximization using a different objective function, and then a combination of various soft and hard product deadline constraints are used. The purpose of including these case studies is to demonstrate the flexibility of this formulation. The objective function as well as constraints can easily be adjusted to realistically reflect various processes as well as to accommodate varying optimization goals.

All case study examples involve a process characterized by n units operating in parallel and separated by $n-1$ intermediate storage tanks. "Batches" of separate feed state materials exist upstream of the first processing unit, and all n processing units operate at a constant and continuous flow rate and these flow rates are product-dependent. Each batch of feed state material is associated with one final product state material. The formulation presented is able to handle situations in which a final product state material is comprised of multiple feed state materials; however, no such cases are considered in the case studies.

4.3.1 Makespan Minimization

Two case studies are now examined to demonstrate the functionality of the optimization formulation presented as well as to demonstrate the benefit of having a flexible maintenance "window".

The first set of case studies examined has a 2-unit 1-buffer process configuration, as illustrated by Figure 4.1. A total of ten tasks (or five jobs) are required to be scheduled in this case study. Based on the programming formulation, this means that tasks 1 through 5 are upstream processing tasks while tasks 6 through 10 are the corresponding downstream processing tasks. In addition, one maintenance event of 8 hours is required on both the upstream and downstream processing units. In case one, each maintenance event is required to begin at $t = 36$ hours (i.e. $T_{mk} = [36, 36]$), while in the second case a 3 time interval "window of opportunity" is given (maintenance must begin between $T_{mk} = [34, 36]$ hours). The time horizon, H, is 72 hours (which is equivalent to 3 days of 24 hour operation). As depicted by Figure 4.2, there are 15 states overall, with 3 for each "job".

Tables 4.2 and 4.3 provide further modelling parameters specific to this case study. Table 4.2 details what family the task belongs to (important for sequence-dependent cleaning), which unit each task must take place in, how long the task takes, and the batch size.

Task i	Family k	Unit j	Processing Time p_i (hr)	Batch Size $B_{i,j}[t]$ (kg)
1	1	1	5	10
2	1	1	5	12
3	2	1	8	14
4	2	1	8	16
5	3	1	10	18
6	4	2	4	10
7	4	2	4	12
8	5	2	5	14
9	5	2	10	16
10	6	2	10	18

Table 4.2: Case study task information.

Table 4.3 groups the states by threes (since each job encompasses three states), and indicates at what time the final (product) state is required in order to meet its deadline, and how much of it is required. Additionally, this table indicates that while initial states and final states have "unlimited" capacity (capacity is set to a value greater than the largest batch size), the intermediate states are bounded by the finite buffer storage. Lastly, the initial amount of material available for processing is given.

Table 4.4 lists the amount of cleaning time required when switching from a task of family k to a task of family k'. Since tasks belonging to families 1 through 3 can only be completed by equipment unit 1 and those belonging to families 4 through 6 can only be completed by unit 2, there is no cleaning time when switching between these tasks. Also, switching from a task of family k to another task of family k does not require any additional cleaning time in this example.

State s	Corresponding Tasks i	Demand Time $t_{demand,s}$ (hour)	Demand $D_s[t_{demand,s}]$ (kg)	Capacity C_s (kg)	Initial Material in State s $S_s[0]$ (kg)
1	1	-	0	50	10
2	1,6	-	0	25	0
3	6	72	10	50	0
4	2	-	0	50	12
5	2,7	-	0	25	0
6	7	72	12	50	0
7	3	-	0	50	14
8	3,8	-	0	25	0
9	8	72	14	50	0
10	4	-	0	50	16
11	4,9	-	0	25	0
12	9	72	16	50	0
13	5	-	0	50	18
14	5,10	-	0	25	0
15	10	72	18	50	0

Table 4.3: Case study state information.

From Family k	To Family k'					
-	1	2	3	4	5	6
1	0	4	5	-	-	-
2	4	0	5	-	-	-
3	3	4	0	-	-	-
4	-	-	-	0	3	6
5	-	-	-	4	0	5
6	-	-	-	4	3	0

Table 4.4: Case study sequence-dependent cleaning information, in hours.

Table 4.5 presents the continuous extension parameters for the case study. The reader should refer to the discussion associated with equation 4.7 for further explanation. Note that in this case study, no material is produced at the beginning of the first time interval relative to the start of the task (i.e. $\bar{\rho}_{i,s}[0] = 0$). This accurately reflects the fact that material cannot move instantaneously through a processing unit; instead, a

1 time interval delay is enforced before feed material can first enter the intermediate storage unit. For consistency, no material is consumed at the beginning of the last time interval relative to the start of a task (i.e. $\rho_{i,s}[p_i] = 0$). This ensures that the 1 hour time interval delay is available for the last amount of material to be produced.

i	s	p_i	$\bar{\rho}_{i,s}[\theta = 1, \ldots, p_i]$	$\rho_{i,s}[\theta = 0, \ldots, p_i - 1]$
1	1	5	-	1/5
1	2	5	1/5	-
6	2	4	-	1/4
6	3	4	1/4	-
2	4	5	-	1/5
2	5	5	1/5	-
7	5	4	-	1/4
7	6	4	1/4	-
3	7	8	-	1/8
3	8	8	1/8	-
8	8	5	-	1/5
8	9	5	1/5	-
4	10	8	-	1/8
4	11	8	1/8	-
9	11	10	-	1/10
9	12	10	1/10	-
5	13	10	-	1/10
5	14	10	1/10	-
10	14	10	-	1/10
10	15	10	1/10	-

Table 4.5: Case study continuous approximation parameters $\bar{\rho}_{i,s}[\theta]$ and $\rho_{i,s}[\theta]$.

The optimal scheduling results with both the 1 time interval ($T_{mk} = [36, 36]$) and 3 time interval ($T_{mk} = [34, 36]$) "window of opportunities" for maintenance are depicted by Figure 4.3. As evident, the 3 time interval window results in a shorter makespan (52 hours instead of 54 hours). This shortened makespan is made possible by starting the maintenance event on the the first processing unit 2 hours earlier (at t = 34 hours) and on the second processing unit 1 hour earlier (at t = 35 hours), and through a change in the sequence of tasks. The comparison of these two cases highlights the

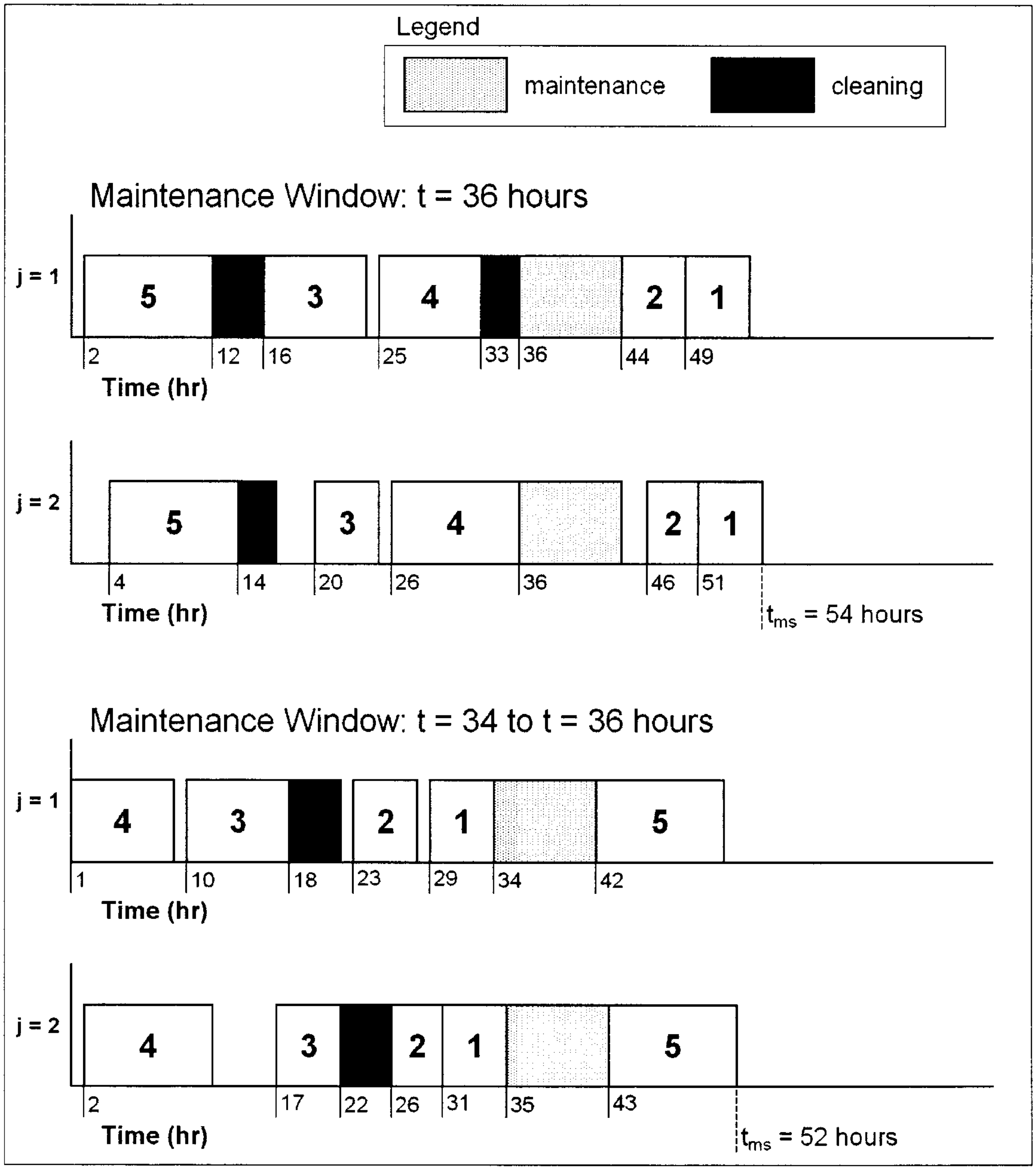

Figure 4.3: Optimized schedules with 1 hour and 5 hour maintenance windows.

importance of selecting an appropriate "maintenance window". Here, by allowing the maintenance events to occur 2 hours and 1 hour earlier, the entire production schedule can be completed 2 hours faster / earlier.

4.3.2 Throughput Maximization

This case study examines the effect of a throughput maximization objective function. Throughput is defined as the sum of the amount of all product state materials over the entire time horizon,

$$throughput = \sum_{s \epsilon D_p} \sum_{t=0}^{H} S_s[t]. \tag{4.24}$$

The objective function then becomes

$$\max_{S_s[t]}(throughput). \tag{4.25}$$

Since maximizing throughput is a result of processing all production jobs as early as possible, it is expected that the optimal schedules arising from this throughput maximization would be quite similar to the makespan minimization results. A comparison of optimal schedules with forced maintenance at $T_{mk} = [36, 36]$ hours for minimized makespan and maximum throughput is provided in Figure 4.4. Note that all constraints and parameters are the same for both cases, the only difference is the objective function.

Table 4.6 is a further comparison of the optimization results. Although both schedules begin maintenance at the same time (36 hours), there are differences in the task sequences and throughput. While the difference in the task sequence has no effect on the makespan, it does effect the throughput; the minimum makespan objective achieves a 7.9% smaller throughput. In conclusion, although the makespan minimization and throughput maximization objectives have a similar logic, they clearly have somewhat different impacts on the overall plant schedule, hence, the objective function should be carefully selected to reflect the true plant priority.

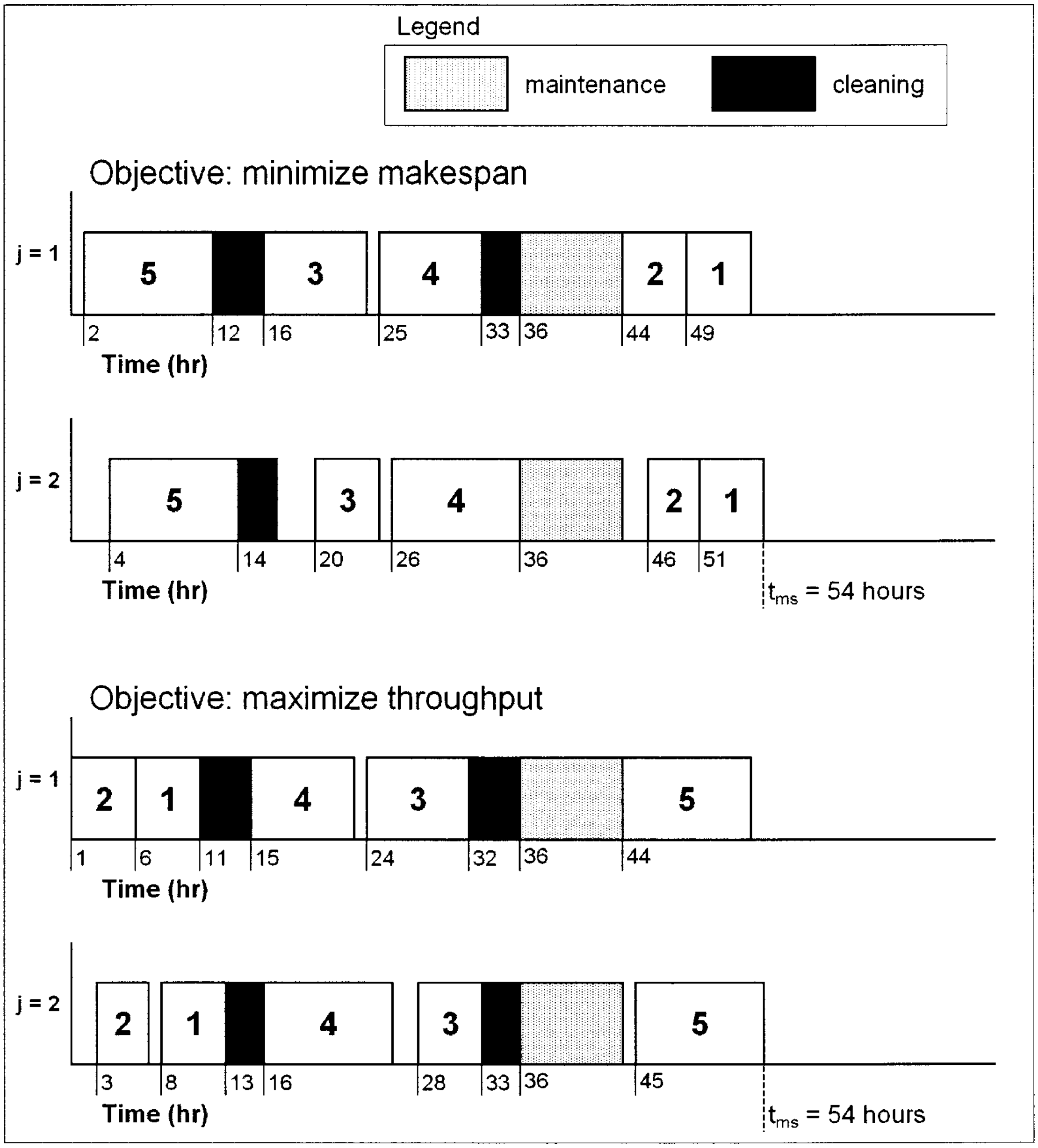

Figure 4.4: Optimized minimum makespan and maximum throughput schedules with forced maintenance at $t = 36$ hours.

	min *makespan*	max *throughput*
Sequence order	5,3,4,2,1	2,1,4,3,5
Makespan (hr)	54	54
Throughput (cumulative kg)	2966	3252
Maintenance start (hr)	36	36

Table 4.6: Comparison of makespan minimization and throughput maximization optimization results.

4.3.3 Soft Product Deadline Constraints

This final objective function variation allows the user to specify soft constraints on meeting product demand deadlines. In this specific example, job J^* is selected as the one and only job with a flexible deadline. Hence, the final product state associated with job J^* (state s^*) is excluded from the hard constraint of equation 4.11. Instead, a product demand deficit for state s^* is now computed:

$$D_{s^*}[t_{demand,s^*}] - S_{s^*}[t] \leq def_{s^*}[t], \qquad t \geq t_{demand,s^*} \tag{4.26}$$

$$def_{s^*}[t] \geq 0 \tag{4.27}$$

$$Def_{s^*} = \sum_{t=t_{demand,s^*}}^{H} def_{i^*}[t] \tag{4.28}$$

Note that inclusion of equation 4.27 ensures that over-production is not penalized.

This deficit is then included in the objective function:

$$\min_{S_s[t],S_{s^*}[t]} (makespan) + I(Def_{s^*}) \tag{4.29}$$

Note that the term I in the objective function is an importance weighting which should be altered to reflect the objective priority as either to minimize makespan, to minimize the product demand deficit, or equally minimize both.

The advantage of including both the makespan and Def_{i^*} terms in the objective function is that the optimizer may select to complete job J^* late, if it will significantly reduce the overall *makespan*. The importance weighting, I, may be adjusted to reflect their comparative priority in the objective function.

Figure 4.5 compares three optimized schedules encompassing two production jobs and one maintenance event on two equipment units in series. All schedules required the maintenance event to begin at $T_{mk} = [15, 15]$ hours, and enforced a hard constraint deadline of $t_{demand,3} = 48$ hours for job 1. The Schedule A model enforced a hard constraint on job 2 to be complete by $t_{demand,6} = 48$ hours, while Schedule B enforced a hard deadline of $t_{demand,6} = 24$ hours on job 2, and Schedule C enforced a soft deadline of $t_{demand,6} = 24$ hours on job 2. As evident, Schedules A and C are identical. In Schedule C, a small value of the importance weighting ($I = 0.1$) placed on the product deficit has the effect of minimizing the makespan in compromise of missing the job 2 deadline. When this importance weighting is made larger ($I \geq 0.2$), Schedule C then becomes identical to Schedule B instead.

For implementation, the selection of the appropriate value of I is directly affected by overall plant economics. Insight into likely economic consequences of late job completion (i.e. financial penalty enforced by the customer, loss of business from customer, and so forth) is required.

If no such economic insight exists, the value of I could be completed by trial-and-error. Observing the resulting optimal schedules from a variety of different values of I would likely be an easy method to obtain an appropriate I (and therefore an appropriate schedule) for cases where one or more product deadlines are flexible. For example, comparison of Schedules A through C in Figure 4.5 may allow for a quick decision on whether or not a 5 hour reduced makespan is worth a compromised 4 hour late completion of job 2.

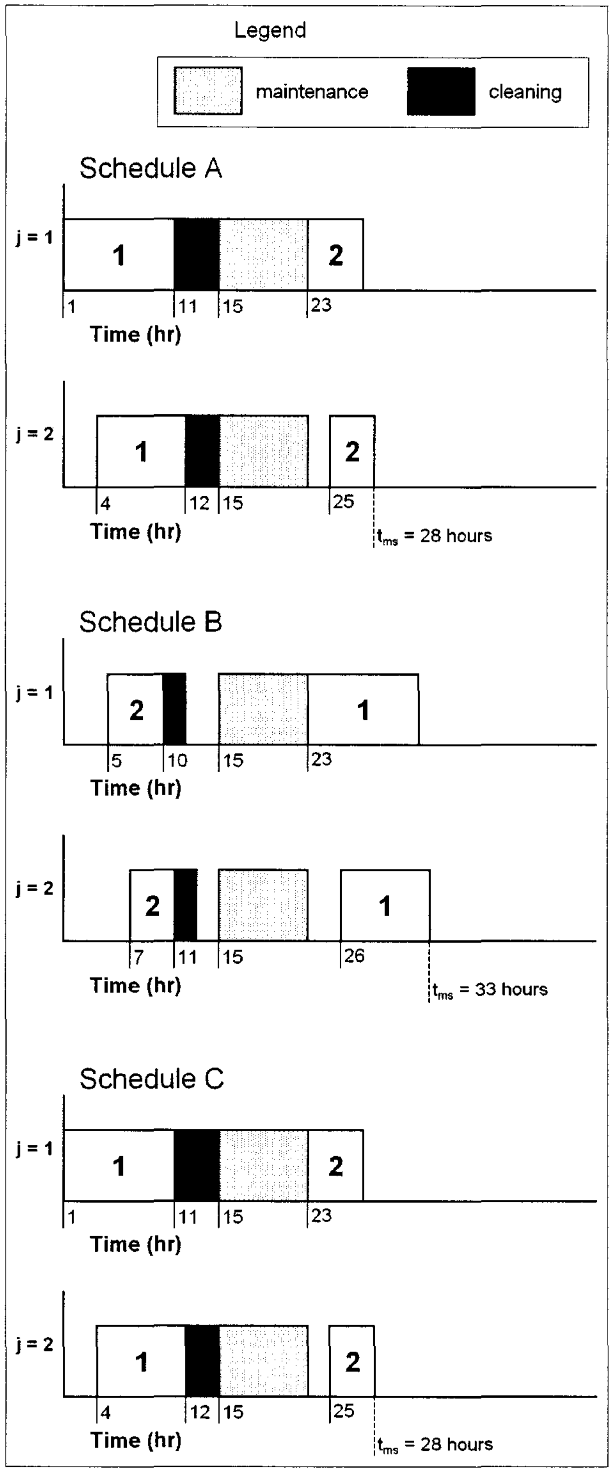

Figure 4.5: Schedule A: Hard demand deadline of t = 48 hours for job 2. Schedule B: Hard demand deadline of t = 24 hours for job 2. Schedule C: Soft demand deadline of t = 24 hours for job 2.

4.4 Continuous Process Formulation

Although the original formulation, based on Kondili *et al.* (1993), did include extensions to approximate continuous process behaviour, this next formulation attempts to more cleanly and more accurately capture continuous process characteristics and dynamics. It should be noted that in this discussion, the term continuous refers the processing characteristics, not the formulation of time (which is discrete).

The purpose of revising the original formulation to a continuous processing form is to allow for job interruption for maintenance (when beneficial). Although the batch formulation based on the work by Kondili *et al.* (1993) would not require significant modification to allow for such job-splitting, significant modification would be required to ensure that processing rates become batch size-dependent. For the type of plant under consideration, the *processing rate* of the unit is constant for a particular feed amount. Thus, if the total amount of material to be processed is split, the processing time would need to be accordingly adjusted to maintain the same, fixed processing rate. As it currently exists, each task is assigned a single *processing time*. Allowing tasks to be split into smaller sub-tasks would require the specification or calculation of an appropriate processing time for every possible sub-task size. Furthermore, the complication of not knowing the number of tasks to be scheduled (as this would be dependent on how many sub-tasks are utilized) would be difficult to handle. The continuous formulation is hence presented as a means of handling job-splitting more cleanly than the batch formulation.

Figure 4.6 demonstrates the advantage of this revised continuous formulation. Schedule A is the optimal resulting schedule of both the batch and the continuous formulations when both job 1 and job 2 are given hard deadline constraints of $t_{demand,s} = 144$ hours. The resulting optimal schedules of the batch and continuous formulations when the job 2 deadline is moved up to $t_{demand,s} = 50$ hours are Schedules B and C

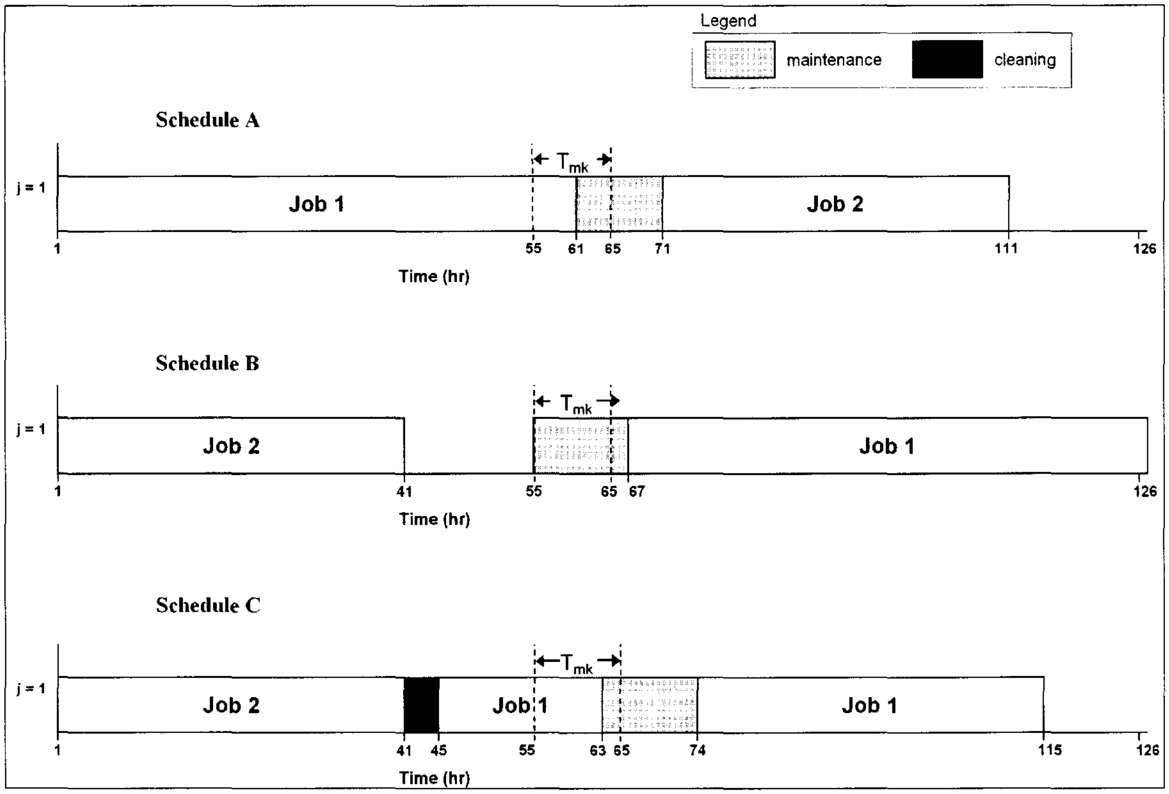

Figure 4.6: Schedule A: Optimal batch (and continuous) formulation schedule with a job 2 deadline of $t = 144$ hours. Schedule B: Optimal batch formulation schedule with a job 2 deadline of $t = 50$ hours. Schedule C: Optimal continuous formulation schedule with a job 2 deadline of $t = 50$ hours.

respectively. Both Schedules B and C have longer makespans than Schedule A - this is unavoidable due to the specified maintenance opportunity window ($T_{mk} = [55, 65]$ hours). The important difference to observe is the difference in makespans between Schedule B and Schedule C. By allowing for job interruption (via the continuous formulation), Schedule C has a makespan 11 hours shorter than Schedule B.

4.4.1 Production Scheduling

The original allocation variable $W_{i,j}[t]$ and equation 4.1 are still used in this formulation; the difference is that $W_{i,j}[t]$ now indicates whether task i is *active* on unit j in time interval t, rather than whether task i begins. Constraint 4.1 is repeated below for completeness:

$$\sum_{i \epsilon I_j} W_{i,j}[t] \leq 1 \qquad \forall j,\, t \qquad W_{i,j}[t] \,\epsilon\, \{0,1\} \tag{4.30}$$

This constraint prevents more than one task from being active on the same unit at the same time.

The material storage capacity constraints are as in Section 4.2:

$$0 \leq S_s[t] \leq M \qquad \forall\, s \,\epsilon\, D_f,\, s \,\epsilon\, D_p,\, t \tag{4.31}$$

$$0 \leq S_s[t] \leq V_s[t] C_s \qquad \forall\, s \,\epsilon\, D_i,\, t \tag{4.32}$$

$$\sum_{s \epsilon D_i} V_s[t] \leq 1 \qquad V_s[t] \,\epsilon\, \{0,1\} \qquad \forall\, t. \tag{4.33}$$

The product demand constraints, as before, are

$$S_s[t_{demand,s}] \geq D_s[t_{demand,s}] \qquad \forall\, s \,\epsilon\, D_p. \tag{4.34}$$

The continuous form of the mass balance now becomes:

$$S_s[t+1] = S_s[t] + \sum_{i\epsilon\bar{T}_s}\sum_{j\epsilon K_i}\bar{\rho}_{i,s}\Delta B_{i,j}W_{i,j}[t] - \sum_{i\epsilon T_s}\sum_{j\epsilon K_i}\rho_{i,s}\Delta B_{i,j}W_{i,j}[t] \qquad \forall\, s,\, t \tag{4.35}$$

The constant parameter, $\Delta B_{i,j}$, represents the mass of material that is processed for task i in unit j over one time interval, and is related to the processing rate of task i through unit j, $F_{i,j}$ (mass/time), through

$$\Delta B_{i,j} = F_{i,j}\Delta t \qquad \forall\, i,\, j \tag{4.36}$$

$F_{i,j}$ is assumed to be known, thus $\Delta B_{i,j}$ may be computed beforehand. We note that in equation 4.35, $S_s[t]$ represents the amount of state s at the beginning of time period t.

In addition to the mass balance, the following constraint ensures that material consumed from the state s at the beginning of time interval t is in fact available at the beginning of time interval t:

$$\sum_{i\epsilon T_s}\sum_{j\epsilon K_i}\rho_{i,s}\Delta B_{i,j}W_{i,j}[t] \leq S_s[t] \qquad \forall\, s,\, t \tag{4.37}$$

For comparison purposes, in Figure 4.6, $F_{i,j}$ was given a value of 1 kg/hr for all i and j, and the amount of materially initially in each feed state in the continuous formulation was set equal to the batch processing time. This ensured that the total processing time for the continuous formulation was equal to the batch processing time.

Similarly to the original mass balance, a normalization (equation 4.38) must be included for the $\bar{\rho}_{i,s}$ and $\rho_{i,s}$ terms.

$$\sum_{s \epsilon S_i} \rho_{i,s} = \sum_{s \epsilon \bar{S}_i} \bar{\rho}_{i,s} = 1 \qquad \forall\, i \tag{4.38}$$

Sequence-dependent cleaning is enforced in this formulation almost the same way as previously:

$$\sum_{i' \epsilon I_j^{(k')}} \sum_{t'=t}^{t+\tau_{j,k,k'}} W_{i',j}[t'] \leq M\Big(1 - W_{i,j}[t]\Big) \qquad \forall\, j,\; i \,\epsilon\, I_j^{(k)},\; t \tag{4.39}$$

$$\sum_{i' \epsilon I_j^{(k \backslash i)}} \sum_{t'=t}^{t+\tau_{j,k,k}} W_{i',j}[t'] \leq M\Big(1 - W_{i,j}[t]\Big) \qquad \forall\, j,\; i \,\epsilon\, I_j^{(k)},\; t \tag{4.40}$$

$$W_{i,j}[t+2] \leq W_{i,j}[t+1] - W_{i,j}[t] + 1 \qquad \forall\, i,\; j,\; t \tag{4.41}$$

Here, the t' summation no longer includes the batch processing time p_i (as it no longer exists). Hence, equations 4.12, and 4.13 are now replaced by equation 4.39, and 4.40. Additionally, a time delay for switching between the same product is now included. Equation 4.41 has the logical effect of: if $W_{i,j}[t] = 1$ and $W_{i,j}[t+1] = 0$ then $W_{i,j}[t+2] = 0$. This delay should be less severe than switching between products within the same family, and since equation 4.41 has the effect of enforcing a 2 time period delay between the same task, all $\tau_{j,k,k}$ and $\tau_{j,k,k'}$ should be greater than 2 time periods.

4.4.2 Extension to Maintenance Scheduling

All five maintenance extension constraints are included here, with only a slight variation to equation 4.17 since it had included the batch processing time. The remaining four equations are included again for completeness, even though they remain unchanged from section 4.2.2.

To ensure that maintenance cannot occur while equipment unit j is processing task i, equation 4.17 is replaced by the following:

$$M_j[t] \leq 1 - W_{i,j}[t] \qquad \forall j,\, i \in I_j,\, t. \tag{4.42}$$

Next, to enforce that a second maintenance task may not begin until the first maintenance task is complete, the following constraint is included:

$$\sum_{t'=t}^{t+T_{j,m}-1} \sum_{m=1}^{M_{tot,j}} x_{j,m'}[t'] - 1 \leq M(1 - x_{j,m}[t]) \qquad \forall j,\, m \qquad t = 1, \ldots, H - T_{j,m} + 1. \tag{4.43}$$

The number of maintenance events that occur on unit j over a given time horizon are defined, as before, by:

$$\sum_{t=1}^{H} \sum_{m=1}^{M_{tot,j}} x_{j,m}[t] = M_{tot,j} \qquad \forall j. \tag{4.44}$$

The following constraint ensures that a consecutive maintenance period of at least $T_{j,m}$ time intervals occurs:

$$\sum_{t'=t}^{t+T_{j,m}-1} M_j[t'] \geq T_{j,m}\, x_{j,m}[t] \qquad \forall j,\, m,\, t = 1, \ldots, H - T_{j,m} + 1. \tag{4.45}$$

The following constraint requires a maintenance task m to begin within the set of consecutive time intervals, T_{mk}:

$$\sum_{t \in T_{mk}} x_{j,m}[t] \geq 1 \qquad \forall j, m. \tag{4.46}$$

The objective function remains unchanged, which for makespan minimization is,

$$\min_{W_{i,j}[t]} T_{ms}. \tag{4.47}$$

However, there is a slight change in the calculation of makespan:

$$T_{ms} \geq W_{i,j}[t]t \qquad \forall i, j, t. \tag{4.48}$$

4.5 Continuous Process Case Studies

4.5.1 Makespan Minimization

The optimal schedule results for two case studies are presented as Figure 4.7; the problem parameters are identical with the exception of the maintenance window. Both cases require a 10 hour maintenance event, however, Schedule A allows this event to begin between $T_{mk} = [20, 22]$ hours while Schedule B is allows this event to begin between $T_{mk} = [19, 23]$ hours. Tables 4.7 through 4.9 detail further case study information. Additionally Δt is given a value of 1 hour.

As evident, Schedule A has a makespan (46 hours) that is two hours longer than Schedule B (44 hours). This is a result of the unit 2 maintenance event being allowed to begin at t = 23 hours in Schedule B. This extra one hour time delay (in comparison with Schedule A) is enough to allow both jobs 1 and 2 to be completed before maintenance begins. Furthermore, Schedule B requires only one cleaning event, since

Task i	Family k	Unit j	Job Size $B_{i,j}[t]$ (kg)	Flow Increment $\Delta B_{i,j}$ (kg)
1	1	1	10	1
2	2	1	8	1
3	3	1	12	1
4	4	2	10	1
5	5	2	8	1
6	6	2	12	1

Table 4.7: Case study task information.

From Family k	To Family k'					
-	1	2	3	4	5	6
1	2	3	4	-	-	-
2	5	2	3	-	-	-
3	4	5	2	-	-	-
4	-	-	-	2	2	2
5	-	-	-	2	2	2
6	-	-	-	2	2	2

Table 4.8: Case study sequence-dependent cleaning information, in hours.

State s	Corresponding Tasks i	Demand Time $t_{demand,s}$ (hour)	Demand $D_s[t_{demand,s}]$ (kg)	Capacity C_s (kg)	Initial Material in State s $S_s[0]$ (kg)
1	1	-	0	50	10
2	1,4	-	0	15	0
3	4	48	10	50	0
4	2	-	0	50	8
5	2,5	-	0	15	0
6	5	48	8	50	0
7	3	-	0	50	12
8	3,6	-	0	15	0
9	6	48	12	50	0

Table 4.9: Case study state information.

maintenance coincides with when the next cleaning event would otherwise occur. Schedule A on the other hand requires two cleaning events - between jobs 1 and job 2, then again between jobs 2 and job 3. This is a result of job 2 being split into two partial jobs - one before and one after the maintenance event. This job splitting should not, however, be considered a disadvantage, and rather, an advantage. In comparison with Schedule B, Schedule A attains a makespan of only two hours longer - which is a very minor difference considering that the batch formulation (which does not allow job splitting) would otherwise have a makespan of 51 hours (5 hours longer than Schedule A, and 7 hours longer than Schedule B), from a job 3 - maintenance - job 1 - job 2 sequence.

4.5.2 Inventory Minimization

A valid concern in most manufacturing systems with intermediate product storage is to monitor, and often minimize, the amount of material in the buffer. As examined in Chapter 3, it is sometimes desirable to maintain a certain amount of material in the buffer, while other times the priority is to minimize intermediate product (due to storage costs, and otherwise available capital if this material was processed into the final product). In this section, the formulation is modified to reflect the possibility of material accumulation in the buffer, and then the objective function is modified to minimize such accumulation.

A simple way of ensuring significant buffer accumulation is to alter the processing rates ($F_{i,j}$) of the upstream and downstream units. Previously, it was assumed that these rates were equal, but here, the downstream processing rate ($F_{i,2}$) is set to half the value of the upstream processing rate ($F_{i,1}$). Without any constraint modifications, the subsequent values of $\Delta B_{i,1} = 1$ kg and $\Delta B_{i,2} = 0.5$ kg are simply inserted into the material balance of equation 4.35, which has been repeated below.

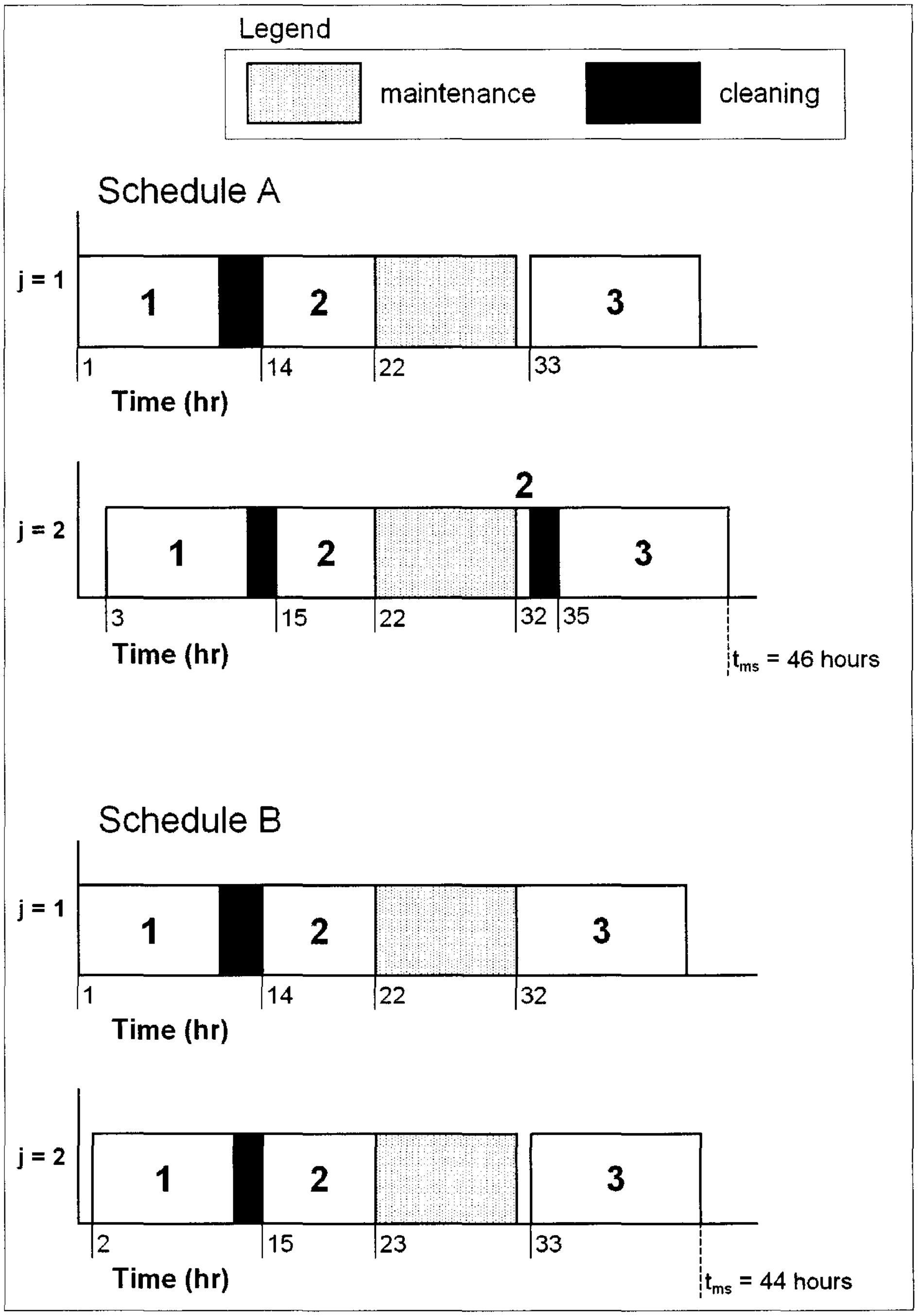

Figure 4.7: Schedule A: Maintenance window of t = 20 hours to t = 22 hours. Schedule B: Maintenance window of t = 19 hours to t = 23 hours.

$$S_s[t+1] = S_s[t] + \sum_{i \epsilon \bar{T}_s} \sum_{j \epsilon K_i} \bar{\rho}_{i,s} \Delta B_{i,j} W_{i,j}[t] - \sum_{i \epsilon T_s} \sum_{j \epsilon K_i} \rho_{i,s} \Delta B_{i,j} W_{i,j}[t] \qquad (4.49)$$

$$\forall\, s,\, t$$

The cumulative amount of material stored in the intermediate buffer is then calculated according to:

$$inventory = \sum_{s \epsilon D_i} \sum_{t=0}^{H} S_s[t]. \qquad (4.50)$$

This amount is then minimized as the objective function:

$$\min(inventory). \qquad (4.51)$$

A small case study requiring one job and one maintenance task to be scheduled is sufficient to demonstrate the effect of altering the objective from makespan minimization to intermediate inventory minimization. Figures 4.8 and 4.9 give optimization results for three different objectives: first for inventory minimization, second for makespan minimization, and thirdly for a two-tiered optimization which first minimizes makespan and then minimizes inventory while maintaining the first objective. In all cases, maintenance may begin between the time interval $T_{mk} = [8, 12]$ hours, and the job to be scheduled must process 10 kg of material. The upstream process rate ($F_{i,1}$) is 1 kg/hr while the downstream process rate ($F_{i,2}$) is 0.5 kg/hr.

Figure 4.8 displays the cumulative amount of material stored in the buffer tank throughout the time horizon for all three optimization objectives. The inventory minimization objective clearly results in the lowest inventory use, while the makespan minimization objective results in the highest inventory use. The two-tiered objective

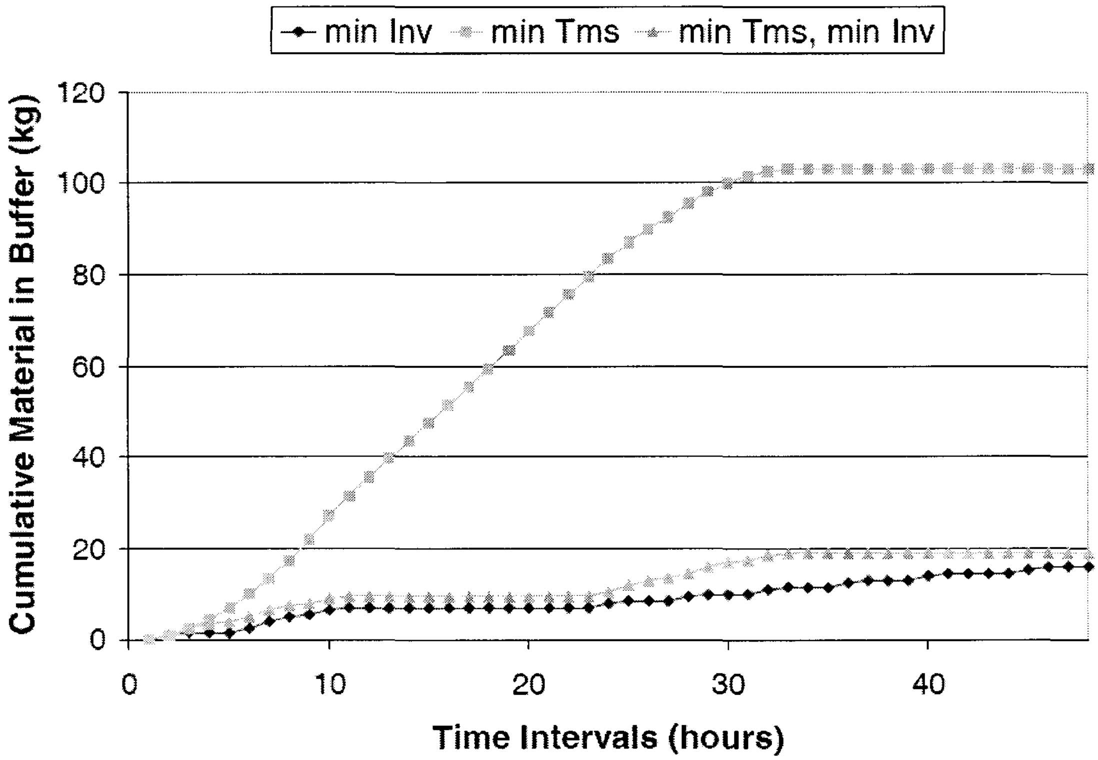

Figure 4.8: Cumulative material storage in intermediate buffer for 3 different objectives: minimize inventory; minimize makespan; tier 1 minimize makespan, and tier 2 minimize inventory.

appears to be very effective, as the inventory use is reduced by more than 80%, while the makespan is still minimized.

The on/off switching behaviour displayed in Figure 4.9 is a direct impact of the inventory minimization objective, especially since in this case study, the downstream processing rate is half of the upstream processing rate. It may then be desirable (or required) to include additional constraints to the model to restrict frequent on/off switching of processing units when executing this objective function. This could be incorporated into the model by means of an additional tier which minimizes variation in the term $W_{i,j}[t]$ over the entire horizon.

A further summary of the three varied objectives is provided by Table 4.10. In these particular case studies, the two-tiered objective provides a good compromise between the two one-tiered objectives: the makespan remains 13 hours shorter than the minimized inventory case, and the cumulative inventory remains 84 kg smaller than the minimized makespan case. The advantage of having a selection of objectives to select from is to accurately reflect realistic priorities for any particular manufacturing plant. For example, a plant with very high intermediate storage costs has the option to select the inventory minimization objective. Additionally, the two-tiered objective could be reversed to first minimize inventory and then minimize makespan.

	min inventory	min makespan	min makespan then min inventory
inventory (kg)	16	103	19
makespan (hr)	46	33	33

Table 4.10: Comparison of inventory and makespan minimization objectives.

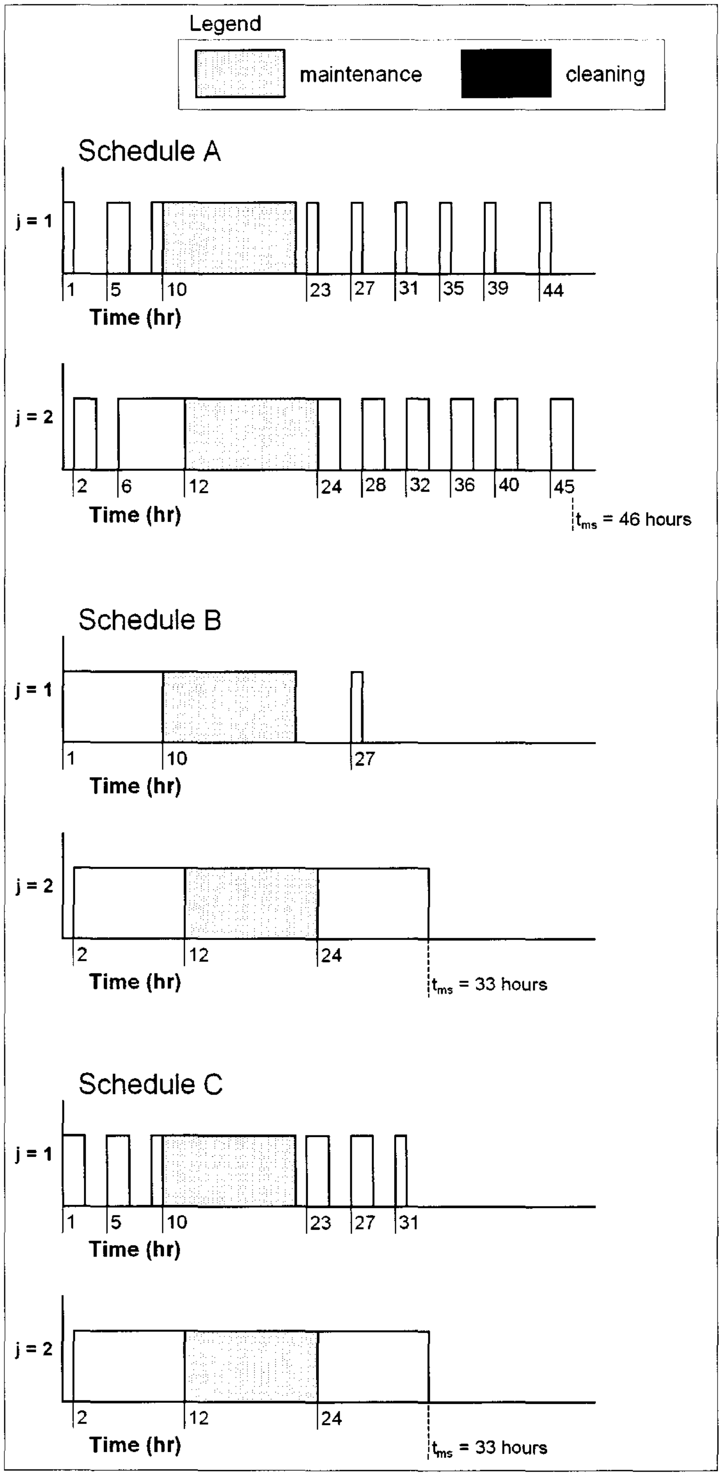

Figure 4.9: Optimal schedule for 3 different objectives. Schedule A: minimize inventory; Schedule B: minimize makespan; Schedule C: tier 1 minimize makespan, and tier 2 minimize inventory.

4.6 Computational Issues

Sufficiently Large Big M Constraints

Several constraints throughout the formulation are of the form

$$x[t] \leq M(y[t]) \tag{4.52}$$

$$x[t] \leq M(1 - y[t]) \tag{4.53}$$

where M is a "sufficiently" large integer. In an effort to improve solution times, the selection of the integer M was examined. It is generally known in the optimization community (Rardin, 1991) that selecting the smallest valid value of M will result in a stronger linear program relaxation. Although a correct integer solution can be obtained from any sufficiently large value of M, Rardin (1991) suggests that it is a "better" choice to select the smallest M, as a stronger relaxation more closely approximates the true model, which results in a number of desired properties including: quicker infeasibility detection, obtaining of sharper bounds, higher chance of detecting an optimal solution, and rounding more easily. Hence, throughout the formulation, all big M values have been minimized.

Batch vs Continuous

Due to the alterations made to move the formulation from batch processing to continuous processing, noticeable differences were found in computation times of the "same" problem. Sequence-dependent cleaning information is provided in Table 4.11. Table 4.12 summarizes the parameters and resulting computation times for the "same" problem in both formulations. The term "same" is put in quotations, as the different formulations do not allow for all of the exact same parameter specifications. In all cases, the processing time (whether specified by p_i in the batch formulation or by $F_{i,j}$, $\Delta B_{i,j}$, and Δt in the continuous formulation) for job 1 is 5 hours, job 2 is 8 hours, and job 3 is 4 hours. Also, a 10 hour maintenance event is required to occur at the specified time interval, and the time horizon is 48 hours.

From Family k	To Family k'					
-	1	2	3	4	5	6
1	2	3	3	-	-	-
2	2	2	2	-	-	-
3	4	3	2	-	-	-
4	-	-	-	2	2	2
5	-	-	-	2	2	2
6	-	-	-	2	2	2

Table 4.11: Case study sequence-dependent cleaning information, in hours.

T_{mk} (start hour)	Solution Time (sec) Batch	Continuous
$[12, 12]$	1.54	13.85
$[12, 18]$	2.55	17.83
$[12, 24]$	3.25	59.17

Table 4.12: Comparison of batch and continuous formulation solution times.

As evident, the continuous formulation requires a longer solution time. However, the largest problem still only required a solution time of less than one minute, which is expected to fall well within the realistic implementation deadline. Since the intended application of this formulation is for off-line periodic computation, a minimum time of 1 to 2 hours for computation is expected to be available.

Problem Size

The computation time is now examined with increasing problem size (increased number of production jobs to be scheduled) and complexity (larger maintenance start time window, T_{mk}) for the continuous formulation. In all cases, the processing time (specified by $F_{i,j}$, $\Delta B_{i,j}$, and Δt) for job 1 is 5 hours, job 2 is 8 hours, job 3 is 4 hours, and job 4 is 10 hours. Also, a 10 hour maintenance event is required to occur at the specified time interval, and the time horizon is 72 hours. Sequence-dependent cleaning information is provided in Table 4.13.

From Family k	To Family k'							
-	1	2	3	4	5	6	7	8
1	2	3	3	5	-	-	-	-
2	2	2	2	4	-	-	-	-
3	4	3	2	3	-	-	-	-
4	3	4	5	2	-	-	-	-
5	-	-	-	-	2	2	2	2
6	-	-	-	-	2	2	2	2
7	-	-	-	-	2	2	2	2
8	-	-	-	-	2	2	2	2

Table 4.13: Case study sequence-dependent cleaning information, in hours.

T_{mk} (start hour)	No. of Jobs Scheduled	Linear Variables	Binary Variables	Constraints	Solution Time (sec)
[12, 12]	1	433	588	3923	0.10
[12, 12]	2	576	888	4563	0.87
[12, 12]	3	719	1188	5201	7.44
[12, 12]	4	862	1488	5837	786.68
[12, 18]	1	433	836	4605	0.17
[12, 18]	2	576	1136	5311	6.07
[12, 18]	3	719	1436	6015	18.35
[12, 18]	4	862	1736	6717	2658.24

Table 4.14: Comparison of continuous formulation solution time with increasing problem size.

Table 4.14 clearly summarizes the dramatic increase in solution time with wider maintenance windows and with an increased number of jobs to be scheduled. The longest computation time (2658.24 sec = 44 mins, 18 sec) occurred with 4 jobs to be scheduled and with a 6-hour maintenance window. This solution time is still, however, assumed to be acceptable for the intended application, with the same reasoning (off-line application) mentioned previously.

Desktop Computer vs High-Performance Server

To improve computation time, McMaster University Advanced Optimization Laboratory's high-performance computing server was used. Table 4.15 below compares several case study solution times on a typical desktop computer with the high-performance computing server. The specifications of the desktop and server are given below.

- **Desktop -** Intel Pentium 4 CPU, 2.80 GHz, 512 MB of RAM, Microsoft Windows XP Professional, Version 2002, Service Pack 2
- **Server -** Tyan S4881, 8 x AMD Opteron 885 dual-core processors (16 cores available, but a maximum of 4 were used for optimization due to software license restrictions), 64GB RAM, OpenSUSE 10.2 Linux

The same problem as described in the previous section (Problem Size) was used to compare total solution times for various cases when run on the server and when run on the desktop. As shown in Table 4.15, small problems gain little benefit in decreased solution times; however, as the problem size increases, so does the difference in solution times.

Depending on the urgency of the optimization results, it may be advantageous to have access to a high-performance computing server when this formulation is applied in industry. However, the intention of this optimization formulation is not for on-line applications, but rather for off-line, periodic updates. It is expected that this optimization would need to be run once a day at the most, and in enough advanced

T_{mk} (start hour)	No. of Jobs Scheduled	Solution Time (sec) server	desktop
$[12, 12]$	1	0.10	2.41
$[12, 12]$	2	0.87	5.27
$[12, 12]$	3	7.44	64.75
$[12, 12]$	4	786.68	6623.84
$[12, 18]$	1	0.17	2.19
$[12, 18]$	2	6.07	6.27
$[12, 18]$	3	18.35	196.17
$[12, 18]$	4	2658.24	20538.39

Table 4.15: Comparison server and desktop solution times with increasing problem size.

time to allow for at least 1 to 2 hours of computation if necessary. The final row in Table 4.15 reveals that the desktop requires close to 6 hours of computation while the server requires only 3/4 of an hour. Although 6 hours of computation may be unacceptable, it is also unlikely that such a large maintenance window ($T_{mk} = [12, 18]$ hours) would realistically be available.

4.7 Chapter Summary

In this chapter two scheduling formulations were presented, both for the intended application of short-term production and maintenance scheduling. Sequence-dependent cleaning, shared finite intermediate storage, and job-specific deadlines were three features included to accurately reflect realistic manufacturing conditions of the industrial motivator. These formulations were based on the work by Kondili *et al.* (1993), but were extended and modified. The key extension was to include maintenance events in the schedule, and to allow for a window of opportunity for when the maintenance event can and must begin. Modification to more accurately reflect continuous processing (i.e. non-batch) conditions was also a major contribution, and was separated out as the second formulation.

Several case studies were included to demonstrate the proper functionality and robustness of the formulations. The objective function was varied to achieve one (or more) of the following: makespan minimization, throughput maximization, and inventory minimization. In addition, product demand deadlines were shown to be flexible; they can either be implemented as soft or hard constraints.

Comparison of the two formulations revealed the usefulness and advantage to accurately modelling continuous processes as such (rather than as batch), in the possible decreased makespan. Decreased makespans from the continuous formulation are attributed to job-splitting, which was not allowed in the batch formulation.

Finally, a brief discussion on computational issues was included. A modification of several constraints to improve solution time, as well as an examination of solution times in relation to problem size, formulation type, and computer used for optimization was included.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

The discussion below summarizes key findings and contributions of this thesis.

5.1.1 Steady-State Buffer Levels with Failure Uncertainty

A problem formulation was developed to determine the optimal nominal steady-state levels for buffers used to separate processing units. These optimal levels were determined based on insight to the frequency and duration of possible unit failures. A multi-period approach was used to incorporate many possible shutdown scenarios, and induced shutdowns were triggered by a drop in flow rate below a specified minimum. The mixed-integer linear programming formulation was then applied to several case studies.

Several case studies were presented for the process configuration of n units separated by $n-1$ buffer tanks. The goal of determining the economical optimal *nominal* buffer level, based on knowledge of various probable failure scenarios was achieved through the formulation of a dynamic optimization problem posed as multi-scenario MILPs. This formulation was coded in AMPL and solved by CPLEX.

Linear relationships were derived for calculating the optimal steady-state buffer levels for simple 2-unit 1-buffer cases with one failure mode per processing unit. The rising complexity of the remaining test cases (3-unit 2-buffer with purge, 2-unit 1-buffer with failure distributions) prevented such simple explanations, and rather demonstrated the need for an advanced optimization to uncover the optimal nominal buffer levels.

The case studies revealed several key findings. First, optimal nominal buffer levels were often "regions" rather than one unique point. Also, initially non-intuitive results were obtained; optimal regions were sometimes split into two regions located near the minimum and maximum capacities, rather than a one centrally located spot. After examination, however, these results were explainable.

A soft-constraint approach was also presented as an alternate formulation which did not require the use of integer variables. Comparison of two soft-constraint case studies with the original formulation had conflicting results; one case closely resembled the mixed-integer results while the other did not. This discrepancy is attributed to the failure of the soft-constraint approach to accurately weight the importance of a unit shutdown.

Finally, the MILP formulation was applied to a pulp mill process. The MILP formulation was able to handle the increased problem size (seven processing units) as well as a more complex process configuration (recycle stream).

5.1.2 Flexible Maintenance Scheduling

A batch as well as continuous scheduling formulation was presented, both for the intended application of short-term production and maintenance scheduling. Again, MILP was utilized in conjunction with a dynamic optimization framework, and all formulations were implemented in AMPL and solved by CPLEX. Sequence-dependent cleaning, shared finite intermediate storage, and job-specific deadlines were three features included to accurately reflect realistic manufacturing conditions of the industrial motivator. Inclusion of flexible maintenance scheduling was the major contribution of this work.

Several case studies were included to demonstrate the proper functionality and robustness of the formulations. The objective function was varied to achieve one (or more) of the following: makespan minimization, throughput maximization, and inventory minimization. In addition, product demand deadlines were shown to be flexible; they can either be implemented as soft or hard constraints.

Comparison of the two formulations (batch and continuous) revealed the functionality and advantage to accurately modelling continuous processes as such (rather than as batch), in the possible decreased makespan. Decreased makespans from the continuous formulation were attributed to job-splitting, which was not allowed in the batch formulation.

Finally, a brief discussion on computation issues was included. A modification of several constraints to improve solution time, as well as an examination of solution times in relation to problem size, formulation type, and computer used for optimization was included.

5.2 Recommendations for Further Work

This project opens many opportunities for future work - the most interesting of which are summarized below.

5.2.1 Steady-State Buffer Levels with Failure Uncertainty

Further investigation is required in regards to the comparison of the MILP formulation and the soft-constraint formulation. Although it may be difficult to analytically characterize the accuracy of the soft-constraint approximation without implementation, a deeper investigation should provide insights and may reveal conditions under which the approximation would be valid.

Additionally, investigation is needed to determine how to best estimate the relative cost / weighting of a unit shutdown compared to profit associated with product, as the value of this parameter, C_u, has a direct impact on the objective.

5.2.2 Flexible Maintenance Scheduling

An obvious extension of this scheduling work would be to include preventative maintenance policies. Since much research has been conducted into optimizing preventative maintenance schedules, it would be beneficial to use those optimization results for setting flexible maintenance scheduling windows centered around this optimized time. A sensitivity analysis would be extremely useful for assessing the long term impact of flexible maintenance scheduling on a plant (including makespan, equipment wear, and so forth).

In addition, a useful extension would include maintenance crew availability constraints.

Finally, a methodology needs to be determined for accurately selecting the relative importance weighting, I, associated soft product deadlines in comparison with makespan minimization.

List of References

AMARI, S. V. (2006). Bounds on mtbf of systems subjected to periodic maintenance. *IEEE Transactions on Reliability*, **55**(3), 469 – 474.

BALTHAZAAR, A. (2005). Dynamic Optimization of Multi-Unit Systems Under Failure Conditions. Master's Thesis, McMaster University.

BRANDOLESE, M., FRANCI, M., AND POZZETTI, A. (1996). Production and maintenance integrated planning. *International Journal of Production Research*, **34**(7), 2059 – 75.

BUZACOTT, J. AND HANIFIN, L. (1978). Models of automatic transfer lines with inventory banks-a review and comparison. *AIIE Transactions*, **10**(2), 197 – 207.

BUZACOTT, J. AND SHANTHIKUMAR, J. (1993). *Stochastic Models of Manufacturing Systems*. Prentice Hall.

CASSADY, C. R. AND KUTANOGLU, E. (2005). Integrating preventive maintenance planning and production scheduling for a single machine. *IEEE Transactions on Reliability*, **54**(2), 304 – 309.

CERVANTES, A. AND BIEGLER, L. T. (2001). *Encyclopedia of Optimization*, chapter - Optimization Strategies for Dynamic Systems, pp. 216–227.

CHEN, J.-S. (2008). Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan. *European Journal of Operational Research*, **190**(1), 90 – 102.

CHONG, Z. (2006). Dynamic Reoptimization and Control Under Shutdown Conditions. Master's Thesis, McMaster University.

CUTHRELL, J. AND BIEGLER, L. (1987). On the optimization of differential-algebraic process systems. *AIChE Journal*, **33**(8), 1257 – 70.

DEDOPOULOS, I. T. AND SHAH, N. (1995). Optimal short-term scheduling of maintenance and production for multipurpose plants. *Industrial & Engineering Chemistry Research*, **34**(1), 192 – 201.

DUBE, J.-F. (2000). Pulp Mill Scheduling: Optimal Use of Storage Volumes to Maximize Production. Master's Thesis, McMaster University.

Fast, R. B. and Caldwell, E. F. (Eds.) (2000). *Breakfast Cereals and How They Are Made*. American Associate of Cereal Chemists, Inc., St. Paul, Minnesota.

FLOUDAS, C. AND LIN, X. (2005). Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. *Annals of Operations Research*, **139**, 131 – 62.

GILL, P., MURRAY, W., AND WRIGHT, M. H. (1986). *Practical Optimization*. Elsevier Academic Press, London, UK.

GRAVES, G. AND LEE, C.-Y. (1999). Scheduling maintenance and semiresumable jobs on a single machine. *Naval Research Logistics*, **46**(7), 845 – 63.

HUANG, Y., REKLAITIS, G., AND VENKATASUBRAMANIAN, V. (2000). Dynamic optimization based fault accommodation. *Computers and Chemical Engineering*, **24**(2), 439 – 444.

IERAPETRITOU, M. AND FLOUDAS, C. (1998). Effective continuous-time formulation for short-term scheduling. 1. multipurpose batch processes. *Computers and Chemical Engineering*, **37**(11), 4341 – 4359.

KONDILI, E., PANTELIDES, C., AND SARGENT, R. (1993). A general algorithm for short-term scheduling of batch operations. i. milp formulation. *Computers and Chemical Engineering*, **17**(2), 211 – 27.

KU, H.-M., RAJAGOPALAN, D., AND KARIMI, I. (1987). Scheduling in batch processes. *Chemical Engineering Progress*, **83**(8), 35 – 45.

LEE, C.-Y. (1996). Machine scheduling with an availability constraint. *Journal of Global Optimization*, **9**(3-4), 395 – 416.

LEE, E. AND REKLAITIS, G. (1989). Intermediate storage and operation of batch processes under batch failure. *Computers and Chemical Engineering*, **13**(4-5), 491 – 498.

LEIVISKA, K., JUTILA, E., PAAVO, U., AND HEIKKILA, S. (1980). Production control of complex integrated mills. *Computers in Industry*, **1**(4), 225 – 233.

LIAO, C. AND CHEN, W. (2003). Single-machine scheduling with periodic maintenance and nonresumable jobs. *Computers &; Operations Research*, **30**(9), 1335 – 47.

LORENZ, K. J. AND KULP, K. (1991). *Handbook of Cereal Science and Technology*. Marcel Dekker, Inc.

MACROSTY, R. (2000). Optimal Scheduling of an Integrated Batch and Continuous Process System. Master's Thesis, University of Cape Town.

MAH, R. S. (1991). *Chemical Process Structures and Information Flows*. Butterworth-Heinemann.

MARAVELIAS, C. T. AND GROSSMANN, I. E. (2003). New general continuous-time state - task network formulation for short-term scheduling of multipurpose batch plants. *Industrial and Engineering Chemistry Research*, **42**(13), 3056 – 3074.

MENDEZ, C. A., CERDA, 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**(6-7), 913 – 946.

PISTIKOPOULOS, E., THOMAIDIS, T., MELIN, A., AND IERAPETRITOU, M. (1996). Flexibility, reliability and maintenance considerations in batch plant design under uncertainty. Vol. 20, pp. 1209 – 14, Rhodes, Greece.

PISTIKOPOULOS, E., VASSILIADIS, C., ARVELA, J., AND PAPAGEORGIOU, L. (2001). Interactions of maintenance and production planning for multipurpose process plants - a system effectiveness approach. *Industrial and Engineering Chemistry Research*, **40**(14), 3195 – 3207.

PISTIKOPOULOS, E., VASSILIADIS, C., AND PAPAGEORGIOU, L. (2000). Process design for maintainability: an optimization approach. Vol. 24, pp. 203 – 8, Keystone, CO, USA.

QI, X., CHEN, T., AND TU, F. (1999). Scheduling the maintenance on a single machine. *Journal of the Operational Research Society*, **50**(10), 1071 – 8.

RARDIN, R. L. (1991). *Optimization in Operations Research.* Prentice Hall.

SHAH, N., PANTELIDES, C., AND SARGENT, R. (1993). General algorithm for short-term scheduling of batch operations - ii. computational issues. *Computers and Chemical Engineering*, **17**(2), 229 – 244.

SWARTZ, C. (2006). Personal written communication.

THOMAIDIS, T. AND PISTIKOPOULOS, E. (1994). Integration of flexibility, reliability and maintenance in process synthesis and design. Vol. 18, pp. 259 – 63, Graz, Austria.

THOMAIDIS, T. AND PISTIKOPOULOS, E. (1995a). Towards the incorporation of flexibility, maintenance and safety in process design. Vol. 19, pp. 687 – 92, Bled, Slovenia.

THOMAIDIS, T. V. AND PISTIKOPOULOS, E. N. (1995b). Optimal design of flexible & reliable process systems. *IEEE Transactions on Reliability*, **44**(2), 243 – 250.

TOLIO, T., MATTA, A., AND GERSHWIN, S. (2002). Analysis of two-machine lines with multiple failure modes. *IIE Transactions*, **34**(1), 51 – 62.

U.S.E.P.A. (1995). Emission Factor Documentation for AP-42 Section 9.9.2 Cereal Breakfast Food Final Report, U.S. Environmental Protection Agency, Office of Air Quality Planning and Standards Emission Factor and Inventory Group.

VASANTHARAJAN, S. AND BIEGLER, L. (1990). Simultaneous strategies for optimization of differential-algebraic systems with enforcement of error criteria. *CCHE*, **14**(10), 1083 – 100.

VASSILIADIS, C. AND PISTIKOPOULOS, E. (1998a). On the interactions of chemical-process design under uncertainty and maintenance-optimisation. pp. 302 – 7, Anaheim, CA, USA.

VASSILIADIS, C. AND PISTIKOPOULOS, E. (1998b). Reliability and maintenance considerations in process design under uncertainty. Vol. 22, pp. 521 – 8, Brugge, Belgium.

VASSILIADIS, C. AND PISTIKOPOULOS, E. (1999). Chemical-process design and maintenance optimization under uncertainty: a simultaneous approach. pp. 78 – 83, Washington, DC, USA.

VASSILIADIS, C. AND PISTIKOPOULOS, E. (2001). Maintenance scheduling and process optimization under uncertainty. *Computers &; Chemical Engineering*, **25**(2-3), 217 – 36.

VASSILIADIS, C., VASSILIADOU, M., PAPAGEORGIOU, L., AND PISTIKOPOULOS, E. (2000). Simultaneous maintenance considerations and production planning in multi-purpose plants. pp. 228 – 33, Los Angeles, CA, USA.

WINSTON, W. L. (1991). *Operations Research.* PWS-KENT Publishing Company, Boston, Massachusetts.

YANG, D.-L., HUNG, C.-L., HSU, C.-J., AND CHEM, M.-S. (2002). Minimizing the makespan in a single machine scheduling problem with a flexible maintenance. *Journal of the Chinese Institute of Industrial Engineers*, **19**(1), 63 – 66.