SCHEDULING OF SINGLE-STAGE NONCONTINUOUS PROCESSES

BALLA GANESH

NATIONAL UNIVERSITY OF SINGAPORE

2004

SCHEDULING OF SINGLE-STAGE NONCONTINUOUS PROCESSES

BALLA GANESH
(B. Tech. REGIONAL ENGINEERING COLLEGE, WARANGAL)

A THESIS SUBMITTED FOR THE DEGREE OF

MASTER OF ENGINEERING

DEPARTMENT OF CHEMICAL & BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

ACKNOWLEDGEMENTS

I am very much thankful to my research advisor *Prof. I. A. Karimi* for his enthusiasm, encouragement, insight and invaluable suggestions during my research at the *National University of Singapore*. His recommendations and ideas have helped me very much in completing this research project successfully. Foremost, I wish to thank him for providing a friendly environment where I was able to work independently. I also thank him for his motivation and moral support throughout my stay here in Singapore.

I gratefully acknowledge the Research Scholarship and other financial support from the *National University of Singapore*. A special thanks to all my lab mates, especially, Reddy, Suresh, Arul, Mohan and Liu Yu for sharing their knowledge with me. I also wish to thank all my friends both in Singapore and abroad, for their constant encouragement and appreciation.

I am deeply indebted to my *parents* and *sisters* for their boundless love, encouragement and moral support. Finally, I express my sincere and deepest gratitude to my eternal motivator, God, for all his love and blessings.

TABLE OF CONTENTS

SUMMARY

This work focuses on scheduling single stage multiproduct noncontinuous plants with non-identical parallel units/lines. In this, the scheduling of both semicontinuous and batch plants with non-identical parallel units/lines and with multiple orders per product was addressed first. The problem structure requires several constraints concerning the release/ready times of units, the due dates of orders and the sequence-dependent setup times. MILP models were developed for both the scenarios with no due date and with multiple due dates using slot-based and event-based time representations. Then, the performance of all models was evaluated and the slot-based models were compared with the event-based models. The results showed that the slot-based models were better than the event-based models. Especially, Relaxed MIP (RMIP) values were more attractive in the case of the slot-based models. Another interesting result was that the decoupling of tasks from units in a mathematical formulation could not reduce the number of binary assignment variables.

Next, two MILP formulations were developed for scheduling single-stage batch plants with demands at multiple due dates. General practice of batch process scheduling has been scheduling batches individually. So the past work could not solve large size problems, where there was a need to schedule multiple batches in one go to meet the demands of different products. In the proposed models, multiple batches of a product were accommodated in one slot to reduce the number of binary values and hence we could solve large size problems with less computational effort. Both the models differ in the way we assign the number of batches to each time slot. The models showed tremendous improvement over existing work in the literature both in terms of model statistics (variables, constraints and non-zeros) and solution statistics (nodes, iterations and solution times). And our work is very robust with respect to the large positive number *M* (big-*M*) that we use in the model formulation.

Finally, the uncertainties in the photolithography station, one of the important processing stages in semiconductor manufacturing processes, were addressed. Different scenarios were considered in which processing times vary with respect to each scenario. Then the stochastic model was compared with the deterministic model. The results showed that the deterministic model not only under predicted the expected value of objective but also could yield a suboptimal solution in the face of uncertainty.

NOMENCLATURE

ABBREVIATIONS

CPI	Chemical Process Industry
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non Linear Programming
NLP	Non Linear Programming

SYMBOLS

Chapters 3-5

Indices

d	Due date
i	Product
j	Unit
k	Slot
n	Event
p	Product
s	Slot

Sets

FS	Slots that are first in their respective due dates
I_j	Products that can be processed on unit j
J_i	Units that can process product i
T_j	Tasks that can be processed on unit j

Parameters

B_{ij}^{L}	Lower bound for the batch size of product *i* on unit *j*
B_{tj}^{L}	Lower bound for the batch size of task *t* on unit *j*
B_{ij}^{U}	Upper bound for the batch size of product *i* on unit *j*
B_{tj}^{U}	Upper bound for the batch size of task *t* on unit *j*
C_i	Cost of unit amount of delayed product *i*
$CT_{ii'j}$	Changeover time for product *i* to *i'*
D_i	Demand of product *i*
D_p	Demand of product p
DD_d	d^{th} due date
FPT_{ij}	Fixed component of the batch processing time of product *i* on unit *j*
FPT_{tj}	Fixed component of the batch processing time of task *t* on unit *j*
I	Number of distinct products
J	Number of units
K	Number of time slots on each unit
K_{d}	Number of the first few slots that can process products due on or before DD_{d}
M	Large positive number
MPL_{ij}	Minimum production length of product *i* on unit *j*
MPL_{tj}	Minimum production length of task *t* on unit *j*
N_{ij}^{L}	Lower bound for the number of batches of product *i* on unit *j*
N_{ij}^{U}	Upper limit for the number of batches of product *i* on unit *j*

O	Total number of orders
Q_{id}	Total amount of product i required on or before DD_{d}
Q_{pd}	Total amount of product p required on or before DD_{d}
R_{ij}^{U}	Maximum rate of production of product i on unit j
R_{tj}^{U}	Maximum rate of production of task t on unit j
RT_j	Release time of unit j
VPT_{ij}	Variable component of the batch processing time of product i on unit j
VPT_{tj}	Variable component of the batch processing time of task t on unit j
α_{id}	Penalty for unit time delay in delivering product i at due date d

Variables

B_{ijk}	Batch size of product i produced in slot k on unit j
B_{tjn}	Batch size of task t produced in event point n on unit j
D_{id}	Time delay in delivering product i at due date d
D_{td}	Time delay in delivering task t at due date d
DA_{id}	Delayed amount of product i due at DD_{d}
DA_{pd}	Delayed amount of product p due at DD_{d}
MS	Makespan
N_{ijk}	Total number of batches of product i produced in slot k on unit j
$SPCDA$	Sum of product of cost and delayed amount
SWT	Sum of weighted tardiness
TE_{jk}	End time of slot k on unit j
TF_{tjn}	Finishing time of task i on unit j in event point n
TS_{tjn}	Starting time of task i on unit j in event point n

$wv(t,n)$	Binary variable, 1 if task i starts in event point n
$X_{ii'jk}$	Continuous variable which takes 1 if i' is following i on unit j in slot k
Y_{ijk}	Binary variable, 1 if product i is assigned to unit j in slot k
$yv(j,n)$	Binary variable, 1 if unit j is being utilized in event point n

Chapter 6

Indices

i	Product
j	Machine
k	Slot
l	Scenario

Parameters

BS_i	Batch/Lot size of product i
N_{ij}^{U}	Upper limit for the number of batches of product i on unit j
P_i	Number of reentrant wafers
PT_{ijl}	Processing time of product i on machine j in scenario l
ST_{ijl}	Setup time for product i on machine j in scenario l
$w(l)$	probability of scenario l
$wc(i)$	Weight coefficient of product i

Variables

B_{ijkl}	Number of wafers of product i on machine j in slot k in scenario l
N_{ijkl}	Number of batches of product i produced in slot k on unit j in scenario l
SI_{idl}	Storage of product i at due date d in scenario l

TE_{jkl}	End time of slot k on machine j in scenario l
$X_{ii'jk}$	Continuous variable which takes 1 if i' is following i on machine j in slot k
Y_{ijk}	Binary variable, 1 if product i is assigned to machine j in slot k
PRO	Production of wafers

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Noncontinuous Chemical Processes

Chemical manufacturing processes can be classified into two types based on their mode of operation. One is continuous and the other one is noncontinuous. A continuous process or unit is the one that produces product in the form of a continuously flowing stream. There is simultaneous input to and output from a continuous unit. Pumping a mixture of liquids into a distillation column at a constant rate and steadily withdrawing product streams from the top and bottom of the column is an example of a continuous process. Continuous processes prefer one unit for each processing step unless equipment is very large and/or redundancy is required. They are usually run as close to steady state as possible and unsteady state conditions exist during the start up of a process. Continuous units are single-purpose units and generally designed for one product/task. They generally produce high-volume, low-margin commodity products. They use dedicated resources and hence allocation, planning and scheduling are not critical.

The continuous processing has been the most prevalent and sought-after mode in the Chemical Process Industry (CPI). Continuous processes, in most cases, are dedicated to produce a fixed product with little or no flexibility to produce other products. They are of large scale and hence require high inventory costs. So in recent years, the noncontinuous processing has received increased attention to manufacture low-volume products. Noncontinuous chemical processes are divided into semicontinuous and batch processes. A semicontinuous unit is a continuous unit that runs intermittently with starts and stops to produce different products. Transition times/costs are needed between the productions of two different products.

Batch process is a production process in which products are produced in batches. In batch process operation, the feed is charged (fed) into a vessel at the beginning of the process and the product/output is collected sometime later. Batch processing plants are attractive due to their suitability for the production of low-volume, high-value products that are becoming increasingly important with rapid market changes. Most batch processes possess the flexibility to produce multiple products. Batch plants are well suited for producing products requiring similar processing paths and/or complex synthesis procedures as in the case of specialty chemicals such as pharmaceutical, cosmetics, wafer fabrication, food, paints, etc. In addition, they are forgiving in the face of seasonal or uncertain demands and lack of process or product knowledge and offer flexibility in terms of operation.

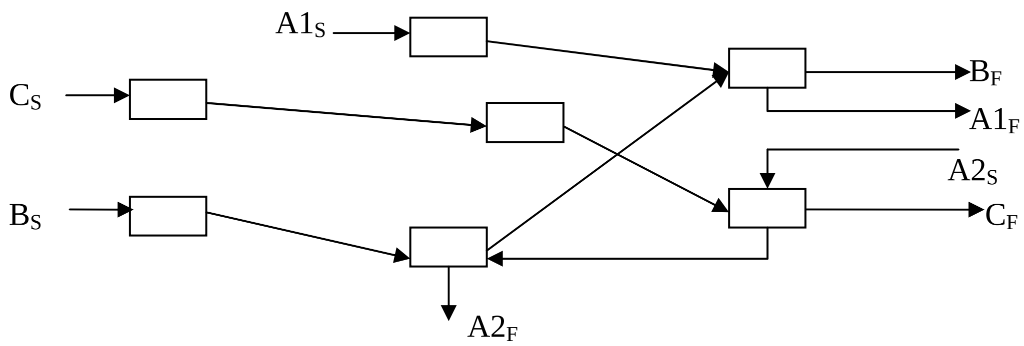
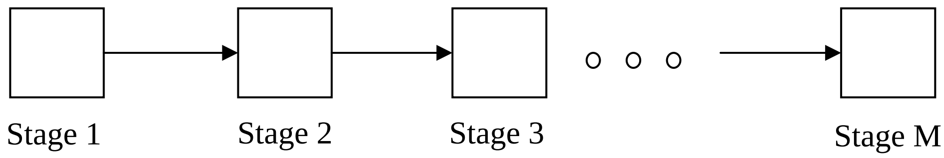


Figure 1.1: Schematic diagram of multipurpose batch plants

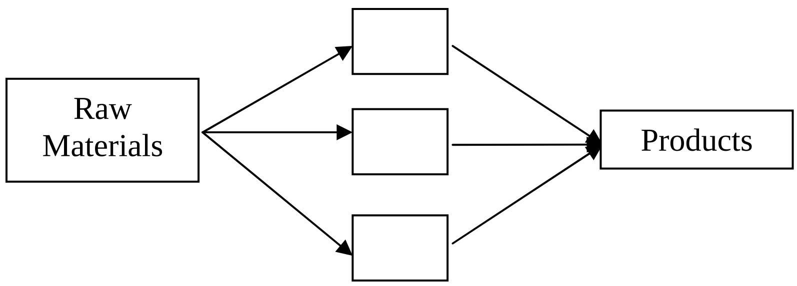
Generally, batch plants have been classified into two types based on the structure of the plants. They are multiproduct and multipurpose. Multipurpose plants are more like a pool of processing units that can be configured into different production lines to produce non-similar products. There are no stages, no fixed configuration. Figure 1.1 illustrates a typical multipurpose plant. Multiple products can be produced at the same time and a product can have multiple routes. In Figure

1.1, product A can be produced in two different ways, shown as A1 and A2. S and F denote the start and the finish states of the product respectively.

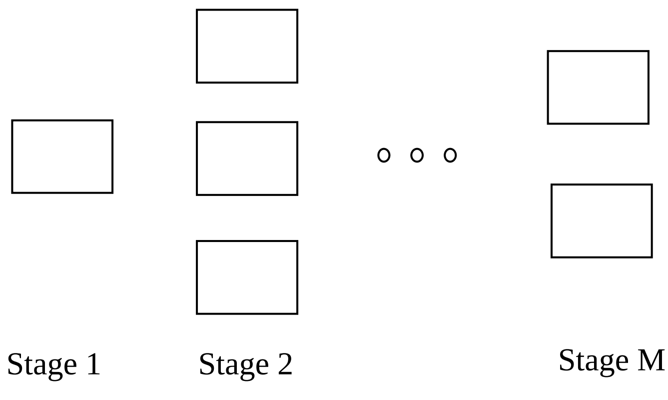
On the other hand, a multiproduct plant is more structured and consists of a series of processing stages, each stage comprising of one or more units. Products produced by multiproduct plants are similar and hence follow the same processing path. Multiproduct plants are classified as serial plants, parallel plants and network plants.



(a) Serial - many stages but only one unit in each stage



(b) Parallel – single stage but more than one unit in parallel



(c) Network – multiple stages and multiple units in parallel in each stage

Figure 1.2: Schematic diagram of multiproduct batch plants

Serial multiproduct plants (Figure 1.2a) consist of more than one processing stage, each stage comprising of only one unit while single stage parallel-unit plants

(Figure 1.2b) consist only one stage with one or more units in parallel. Network multiproduct plants (Figure 1.2c) are hybrid of the above two i.e. consist of more than one processing stage, each stage comprising one or more units in parallel.

The main attraction of noncontinuous plants is their inherent flexibility in utilizing the various resources available for the manufacture of relatively small amounts of different products of more batches within the same production facility. Since sharing of resources (time, equipment, manpower, utilities, raw materials, etc.) to manufacture multiple products is the principal feature of noncontinuous plants, the need for optimization invariably arises both in the design and the operation of such plants. Sophisticated planning and scheduling tools are needed to allow the utilization of resources in a way that takes full advantage of the flexibility of these plants. But what is planning? What is scheduling?

1.2 Planning and Scheduling

Planning and scheduling are methodologies that enable us to run the chemical plants smoothly. Planning and scheduling occur in a wide range of economic activities. They always involve in accomplishing a number of things that tie up various resources for periods of time. Planning horizon typically spans a period until which the complete demand information is available. Generally, it consists of long horizon ranging from months to years. It decides how much amount of products the plant should produce and how long units should run the operation to get desired products. Simply, planning is an economic criterion.

Scheduling, a short-term planning, ranges from weeks to months. It is a methodology that determines the order in which products are to be processed in each of the units so as to optimize some suitable performance criterion. Simply, a scheduling system is a system that makes decisions dynamically about matching activities and resources in order to finish products and projects that require these activities in a timely and high-quality fashion while simultaneously meeting the desired economic criterion. The basic scheduling decisions that are to be made include sequencing of products on units, timing and releasing of units and orders, and exact timings of activities. In addition, decisions regarding resources, forecasting, aggregating and disaggregating also play an important role.

Scheduling is very much important in the entire supply chain network that can be defined as a network of facilities and distribution options. Supply chain management performs the functions of procurement of materials, transformation of these materials into intermediate and finished products, and the distribution of these finished products to customers. Effective coordination of supply chain elements such as ordering, production management that encompasses scheduling, inventory management, distribution management, makes an organization, a profitable one.

1.3 Need for Scheduling

The need for scheduling arises neither from the nature of the processing operations i.e. continuous, semi-continuous, or batch nor is determined by the properties of the processed materials. But the need arises from methods that are used by CPI to allocate resources to products. Current scheduling practices involve the usage of manual, tedious and error-prone spreadsheet tools that result in adhoc procedures. Moreover, the scheduling is done by dedicated staff whose decisions may be limited only to human imagination with the possibility of eluding the optimum. This results in lower utilization of resources available and the productivity suffers even further.

Some plants may use dozens of equipment to produce different types or grades of products. This leads to a myriad of ways in which a plant can be operated, and finding the best operating plan and schedule becomes a challenge. Adding the dimension of time to the above, one has complex combinatorial problems that are impossible to solve optimally using a manual spreadsheet that is widely used by the industry personnel. Clearly, an enormous potential exists for improving the productivity and profitability of chemical plants by means of systematic, computer-aided, decision support tools that use advanced optimization methods. The large variety of such plants with diverse requirements, features and uncertainties has fueled extensive optimization research during the past two decades.

1.4 Scheduling in Noncontinuous Plants

Scheduling noncontinuous plants is very difficult and challenging given the flexibility of such plants. Certain salient characteristics of these plants need to be carefully observed and incorporated in scheduling decisions:

1. A variety of products are produced in batches and several batches are produced simultaneously.
2. Orders arrive at different times and their due dates are usually tight.
3. Highly capital intensive processing and material handling equipment are employed.
4. Processing equipment is functionally versatile.
5. Real-time control of scheduling decisions is required to respond to the dynamic behavior of the system and to attain an effective utilization of resources.
6. Decisions about various manufacturing resources are required to be coordinated in order to exploit the flexible nature of the noncontinuous plants.
7. The release times and due dates of orders, as well as the sequence-dependent setup times and forbidden sequences of production orders and the ready times of units play an important role in scheduling these plants.

Scheduling, in general, involves (a) deciding about the scheduling horizon, (b) establishing suitable objective(s), (c) modeling the characteristics of the plant structure, and (d) employing a suitable solution methodology for the resulting formulation.

The scheduling horizon, in general, is in the range of 4 to 8 weeks and the information on demands of various products decides the horizon. The various scheduling objectives can be classified into two broad categories: (i) product related objectives – tardiness, makespan, flowtime, earliness, productivity and costs; and (ii) resource related objectives – utilization, idle time and costs. Quite often, scheduling decisions are made with more than one objective. But we should select the objective(s) in such a way that it truly reflects the performance of the system or plant. Accurate constraints regarding assigning products to units, sequencing products on units, resource allocations and objective make a mathematical model a complete and useful one.

The solution methodology for scheduling encompasses mainly two approaches: one is mathematical programming and the other one is heuristic approach. Both approaches have advantages and disadvantages. These two methods have been used extensively to solve scheduling problems arising in non-continuous chemical plants. The mathematical programming approach results in either Nonlinear Programming (NLP) or Mixed Integer Linear Programming (MILP) or Mixed Integer Nonlinear Programming (MINLP) formulations. MILP approach is a popular approach for solving scheduling problems as it guarantees global optimal solutions. But mathematical models are computationally expensive and also as the problem size and complexity increase, most mathematical formulations fail to give optimal solutions in reasonable amount of time. On the other hand, heuristic algorithms such as genetic algorithm, tabu search, simulated annealing (SA) etc. are computationally less expensive than mathematical programming but most of the time they give suboptimal solutions. There is little advantage or carry-over from one problem to another. Furthermore, the ability of most heuristic algorithms to give good suboptimal solutions deteriorates rapidly with increasing problem size.

1.5 Research Objective

The main focus of this work is to develop methods for determining optimal production schedules for a fixed set of units that form a stage. A significant body of research work exists in this area with focus on the development of exact and approximate methods to solve scheduling problems. In this work, the focus is on multiproduct single-stage noncontinuous plants. Semicontinuous and batch processes with no due date and multiple due dates are considered. Models are developed using slot-based and event-based formulations and also a comparison is drawn between the two approaches.

Past scheduling work on scheduling batch units addressed the problem in terms of individual batches. With this methodology, it becomes impossible to schedule large number of batches that are waiting before the units for processing. Thus, there is a need to develop a mathematical model that addresses the problem in terms of multiple batches suitable for a unit. So, the objective of this research work is to develop models that are more effective than the previous models existing in the literature in terms of objective value and computational time.

1.6 Outline of the Thesis

In the next chapter, a detailed literature review is presented. In chapter 3, the scheduling problem is discussed. In chapter 4, the short-term scheduling problem with no due date and with multiple due dates is considered. Four different slot-based models are presented separately for semicontinuous and batch processes. In addition, corresponding event-based models are formulated and a comparative analysis is carried out between slot and event based approaches with the help of some illustrative examples.

In chapter 5, scheduling of single stage batch plants with intermediate due dates for the general case of multiple orders per product is presented using a novel approach. Two different MILP formulations are developed. They are compared with the existing models in the literature via some literature examples.

In chapter 6, the application of the above work in semiconductor manufacturing with uncertainties is addressed. Various parameters are considered which are dynamic in nature and the performance of the system is evaluated. Some conclusions are drawn about this work in chapter 7 with some recommendations for future work in the area of scheduling of noncontinuous plants.

Chapter 2

LITERATURE REVIEW

2.1 Time Representation

Time representation is very important while developing mathematical models for scheduling plants. This is because the overall profile of resource utilization is discontinuous. The model has to track such discontinuities within the scheduling horizon i.e. the profile is compared with the resource availabilities to ensure feasibilities. There are two existing approaches for representing time in mathematical formulations to deal with such complexities. One is discrete-time representation and the continuous-time representation is being the other one.

Discrete-time representation was used by researchers (Kondili et al., 1993; Shah et al., 1993) vastly. In discrete-time representation, the scheduling horizon is divided into a number of intervals of equal duration. Events of any type such as the start or end of processing individual batches of individual tasks, changes in the availability of processing equipment and other resources, etc. are only allowed at the interval boundaries. Simply, all the tasks must begin and end at the boundary of an interval. The main advantage of this type of representation is that it facilitates the formulation by providing a reference grid against which all operations competing for shared resources are positioned. Figure 2.1 shows the schematic diagram of discrete-time representation.

The discrete-time representation is used only when the processing times of products on units are constant and, furthermore, the duration of intervals must be equal to the highest common factor of the processing times involved. The assumption of constant processing times is not always realistic, and the length of the intervals might be so small that it either leads to a prohibitive number of intervals, rendering the resulting model unsolvable, or else requires approximations that might compromise the feasibility and optimality of the solution.

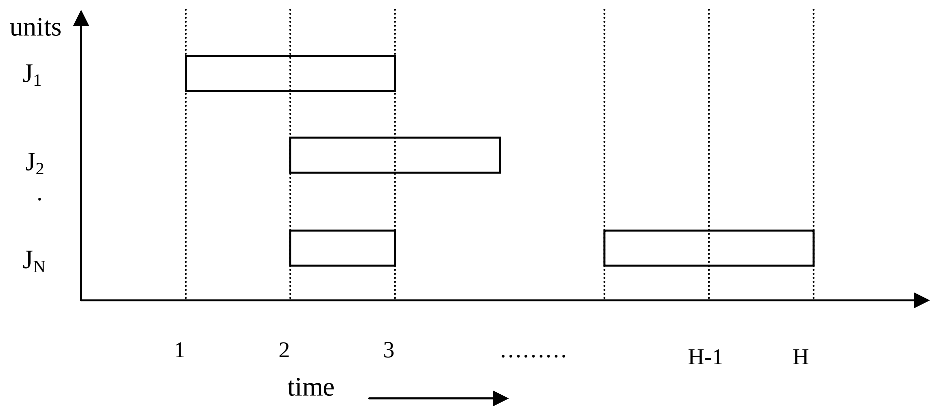


Figure 2.1: Schematic diagram of discrete-time representation

In discrete-time representation, there is a binary variable associated with each interval which indicates whether or not that task is started at the beginning of that interval. Thus time is considered as a discrete variable which can attain the values of the beginning of each interval. So the main difficulty with this representation is that in order to represent a process accurately, we may need to develop a model with a very large number of binary variables. To decrease the number of binary variables, rounding of event times and duration is commonly used. The drawback of rounding is that it is difficult to use such a schedule for process control without ad hoc adjustments because the process control logic requires precise execution times. Furthermore, rounding up can produce infeasible or loose schedules while rounding down can produce infeasible schedules. Another inherent difficulty of discrete-time representation arises in representing continuous processes. A continuous process may start and end somewhere within an equal size interval, not on the interval boundaries. These two limitations are removed by the continuous-time representation. The continuous-time representation accounts for variable processing times and is more realistic than the discrete-time representation. It also requires significantly fewer time intervals and hence leads to smaller problems.

Most of the researchers, generally, have developed continuous-time formulations using any one of the following three approaches. They are sequence-based method, slot-based method and event-based method. As the name suggests, sequence-based method is based on the sequence of products that are processed on units. In this, researchers (Cerda et al., 1997; Mendez et al., 2000; Gupta and Karimi, 2003) assign binary variables for sequencing one product after another and processing those products on units. Mathematical formulations based on sequence-based are, so far, not free of Big-*M* that affects the solution times drastically.

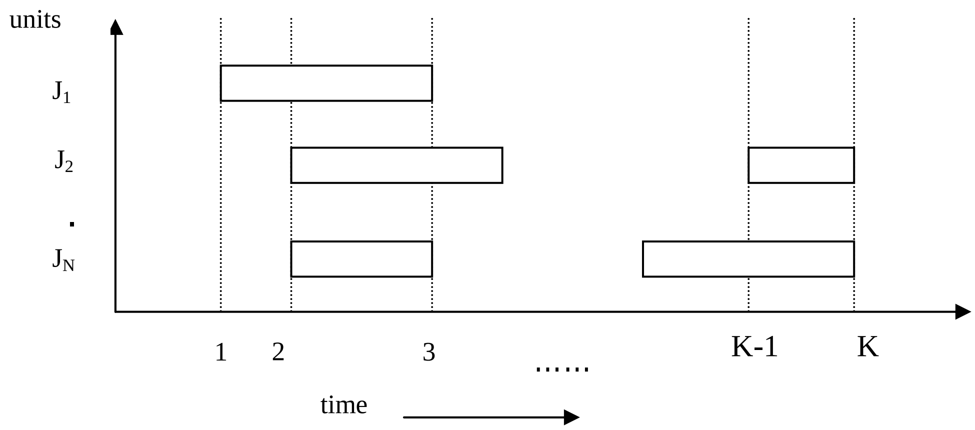


Figure 2.2: Schematic diagram of synchronous time representation-I

The second one, slot-based method is quite effective in developing models based on variable-length time slots (Lim and Karimi, 2003a; Lamba and Karimi, 2002a; Sundaramoorthy and Karimi, 2004). Binary variables are necessary to decide which product should occupy which unit in which slot. There are two ways to represent time in slot-based models. They are synchronous and asynchronous. In both, the time slots need not be identical but each time slot is equal in length for all units in synchronous time representation where as in asynchronous, the length of a slot on a unit need not be the same as on other units. Figure 2.2 and Figure 2.3 represent synchronous time representation.

In Figure 2.2 and Figure 2.3, the time horizon is divided into intervals of unequal and unknown duration, common for all units. In Figure 2.2, each task must start and finish exactly at a time point. But in Figure 2.3, each task must start at a time point but need not finish at a time point. These are few examples for synchronous representation.

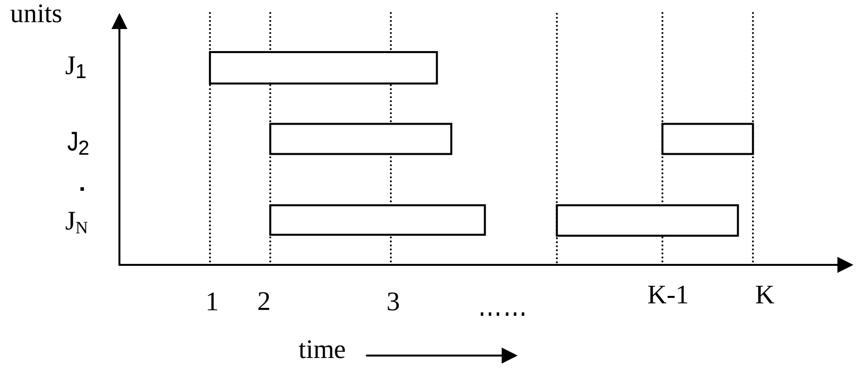


Figure 2.3: Schematic diagram of synchronous time representation-II

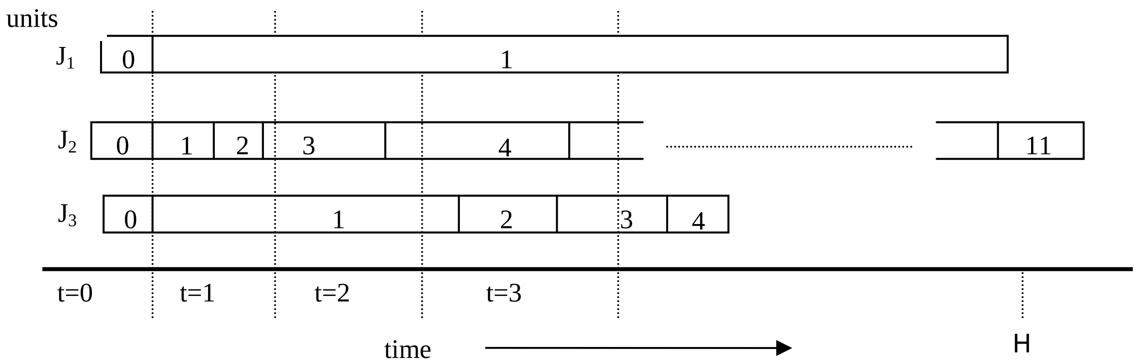


Figure 2.4: Slot design in model M1

In asynchronous representation, time slots are not common for all units. Karimi and McDonald (1997) proposed two asynchronous slot-based models (model M1 and model M2) that differ in defining time slots. Figure 2.4 and Figure 2.5 represent model M1 and model M2 respectively.

Model M1 and Model M2 differ in the design of time slots and how they are assigned to periods. In M1, slots are of arbitrary lengths and are independent of periods. A slot is not confined to be within one or more periods. It may cover one or more periods and can even extend beyond the scheduling horizon. In M2, each time period is divided into a fixed number of slots a priori. Thus, a slot is confined to be within a single period and its length cannot exceed the length of its period. Note that asynchronous time representation almost always needs fewer time slots than synchronous time representation.

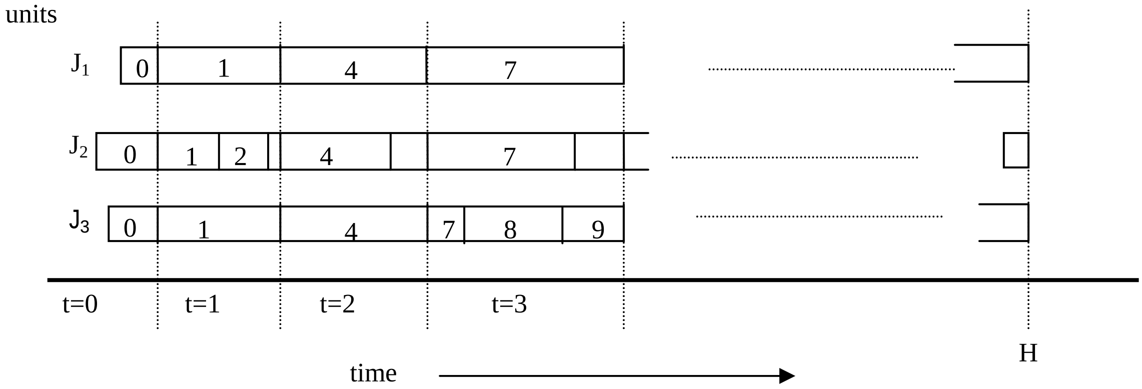


Figure 2.5: Slot design in model M2

The slot-based model requires the initial guess of the number of slots a priori. Though we can define some rules or formulae to estimate number of slots, those rules or formulae may overestimate the number of slots required and hence may increase the number of binary variables.

Ierapetritou and Floudas (1998) introduced event-based formulations. The event-based time representation is purely asynchronous. The basic idea of their work is that it decouples task events from unit events. This is achieved by the consideration of different binary variables to represent the task events and unit events. Maravelias and Grossmann (2003) also implemented the concept of decoupling for short term scheduling of multipurpose batch plants. But the concept of decoupling was questioned by Sundaramoorthy and Karimi (2004). In the event-based formulations too, we have to guess the number of event points a priori.

With this detailed description about time representation, the past work in the area of noncontinuous plant scheduling will be discussed in the next section.

2.2 Past Work

In CPI, especially in noncontinuous process plants, it is essential to enhance process flexibility and production efficiency due to changing market demands and various customer requirements. A wide range of products in the CPI are produced using noncontinuous (batch and semicontinuous) mode of production. This mode of production has long been the accepted procedure for the manufacture of many types of chemicals (specialty chemicals, pharmaceuticals, polymers, bio-chemicals, foods, etc.), particularly those which are produced in small quantities and for which the production processes or demand patterns are likely to change.

The most important feature of batch processes is their flexibility in processing multiple products by accommodating the diverse operating conditions associated with each product. Therefore, in spite of the traditional drive towards continuous production, the batch mode continues to be the only alternative for a number of sectors of the processing industry. As a result, the importance of effective tools for scheduling and planning activities within the CPI has grown with the increasing emphasis on customer satisfaction, reduced inventory, lower manufacturing costs and global operations. To increase process flexibility and profitability, efficient scheduling techniques are needed.

The problem of short-term scheduling of noncontinuous plants seeks to determine the optimal strategy for satisfying the production demands of a variety of products at specific dates and/or at the end of a given production horizon. The short-term scheduling is challenging especially for multiproduct single-stage batch plants, where the production of individual batches, even for the same product, does not follow the same pattern but must be specified according to an overall performance index and is subject to capacity and time constraints. It involves the allocation of equipment and resources to orders and the sequencing of these orders.

Short-term scheduling of multiproduct single-stage batch plants has received considerable attention over the past two decades. Many diverse approaches, mathematical formulations and solution algorithms have been proposed. Scheduling of batch processes was widely discussed in the literature and extensive reviews were given by Reklaitis (1991), Pinto and Grossmann (1998). Many of these problems can be posed as mixed integer optimization problems, since the corresponding mathematical optimization models involve both discrete and continuous variables and a set of equality and inequality constraints to be satisfied.

Cerda et al. (1997) proposed a continuous-time formulation for short term scheduling of a single-stage multiproduct batch plant with parallel units. They used a tri-index binary variable that governs predecessor-successor-unit information to assign the orders to the non-identical production units while taking into account sequence-dependent changeover constraints. To deal with large-size problems, they proposed heuristics, such as pre-ordering of the orders to reduce the number of feasible predecessors for each order. But they considered single order per product which is not general in CPI.

Mendez et al. (2000) used the same modeling approach but employed a different set of binary variables and considered multiple orders per product. Instead of the tri-index binary variables used by Cerda et al. (1997), their formulation was based on two bi-index binary variables, one indicating the predecessor-successor relationship and the other governing the allocation of a product to a unit. In the first phase of their approach, the product batching process is accomplished to minimize the work-in-process inventory while meeting the orders' due dates. In the second phase, the set of batches obtained in the first phase is optimally scheduled with the objective being minimizing tardiness to meet the product orders as close to their due dates as possible. But later, the work by Lim and Karimi (2003a) proved that the formulation of Mendez et al. (2000) resulted in suboptimal solutions.

Hui and Gupta (2001) presented a sequence-based MILP that deals with single order per product. Similar to Mendez et al. (2000), they used three sets of binary variables. For large problems, a pre-ordering heuristic was used to reduce the number of binary variables. But this could not guarantee optimum and also their methodology could not be applied to multiple orders per product.

Lim and Karimi (2003a) proposed a slot-based MILP formulation that deals explicitly with multiple orders per product. Unlike Mendez et al. (2000), their formulation decides both product batches and their schedule in one step. They, in fact, proposed two models. One is a general model that accounts for multiple orders per product and the second one is for the special case of a single order per product. They showed that the former is more effective of the two models as the later resulted in suboptimal solutions. They also addressed the effect of big-*M* on MILP models. According to their work, big-*M* plays a very important role in the model performance and should be avoided if possible while developing mathematical models.

Chen et al. (2002) developed a slot-based MILP model. In their work, the allocation of orders and units to time slots is represented by two sets of binary variables (order-slot and unit-slot). To reduce the size of the model, two heuristic rules were developed. But the time representation considered in their work cannot be applied to multi-stage scheduling problem.

Ierapetritou et al. (1999) presented a continuous formulation for short-term scheduling of batch plants using the event-based method. Their idea was based on the decoupling of task events from unit events. But the major limitation of this formulation was that it required pre-ordering of all of the orders in advance that restricted its application to specific problems. It did not consider the sequence-based setup times that are more common in CPI.

Having discussed the works related to the scheduling of single-stage parallel units operating in the batch mode in detail, the literature concerning the semicontinuous operations of the same needs to be discussed. Short-term scheduling of semicontinuous processes has received less attention compared to batch process scheduling in the chemical engineering literature. Sahinidis and Grossmann (1991) addressed the problem of cyclic multiproduct scheduling on continuous parallel production lines. They developed a MINLP model and applied an exact reformulation technique to linearize it in the space of the integer variables. Pinto and Grossmann (1994) followed the above work and extended it to multistage case. They allowed intermediate storage between stages and the resulting complexities in the mass balance equations were taken care by some algorithms presented in their work.

Karimi and McDonald (1997) developed two slot-based continuous-time models based on different time scale representation. The two models differ in pre-assignment of slots to time periods/due dates. They implemented several realistic situations such as transitions, inventory costs, safety stocks and due dates. Their formulation addresses to solve the problem of a single-stage multiproduct facility with parallel semicontinuous processors. But they did not consider resource constraints.

Lamba and Karimi (2002a) addressed the resource constraints using synchronized time slots that are identical on every unit. Their designation of slots is such that for any change on any production line, a slot change will be triggered on all lines. They considered sequence dependent setup times for each unit with makespan as objective. The same authors proposed a decomposition algorithm in their next publication (Lamba and Karimi, 2002b). The main idea behind this algorithm was to first generate several good, feasible item combinations by repeatedly solving the model with a minimum number of slots and then to compose a schedule using these item combinations.

Recently, Lim and Karimi (2003b) considered resource constraints and other necessary features of a semicontinuous process and developed models that differ in the use of variables and constraints using asynchronous slots. They use “check points” to ensure that resource constraints are always satisfied. By using asynchronous slots, they reduced the number of binary variables and hence computational effort.

As was the case for the batch mode of operation, Ierapetritou et al. (1999) had developed an event-based MILP formulation for the semicontinuous processes. They claimed that their model resulted in less number of binary variables than Karimi and McDonald (1997) using the concept of decoupling. Sundaramoorthy and Karimi (2004) have discussed the concept of decoupling in detail. They showed that decoupling of tasks from units in a scheduling formulation cannot reduce the number of binary assignment variables. In the next section, a brief introduction about our work is presented.

2.3 Research Focus

With the primary objective of short-term scheduling being the determination of the start and end time of operation(s), the representation of time has been a subject of debate among the researchers in this field. The formulations based on discrete-time representation results in large number of binary variables due to the discretization of time and have fallen out of favor. The focus is now on the continuous-time formulations for it results in compact models allowing one to capture the operations accurately. The three most prominent approaches in the continuous-time domain models of short term scheduling are the slot-based approach, the sequence-based approach and the event-based approach. As there have been several reservations among the researchers with regard to the three approaches, there is a long-standing need to compare and analyze the nuances of each of them. In the first part of this work, an effort is made to compare the slot-based and event-based approaches for single-stage multiproduct noncontinuous plants with non-identical parallel units.

Though the scheduling problem involving several identical orders and batches has received considerable attention, still a model that uses less binary variables and accommodates/schedules multiple batches in one slot is missing in the literature. So, in the second part of this work, the main focus is on developing such a model using slot-based approach for scheduling single-stage batch plants with parallel units and products having multiple orders at multiple due dates. The effect of Big-*M* on mathematical models is considered which has received less attention in the literature. Big-*M* constraints are a sort of penalty constraints that bind for the desirable condition and relax for the undesirable condition with the help of a huge penalty (a big positive number). There is no estimate for *M*-value in the mathematical models and hence it results either in suboptimal solutions or poor solution times. In this research, this issue is addressed and it is shown that the models are very robust to *M*-value.

The third part of this work deals with the uncertainties in chemical batch plants which are very common in general. Most of the works in the literature assume the data as deterministic while developing mathematical models. However, in real plants, parameters such as raw material availability, processing times, and market requirements vary with respect to time and are often subject to unexpected deviations. These uncertainties are common and can have undesirable short-term and long-term economic and feasibility implications. Therefore, the consideration of uncertainty in scheduling problems is of great importance in preserving plant feasibility and viability during operations. So an effort is made to develop a MILP formulation to account for these uncertainties in single-stage multiproduct batch plants with reentrancy.

Chapter 3

PROBLEM DESCRIPTION

The problem addressed by Lim & Karimi (2003a) is generalized here. Figure 3.1 shows a schematic diagram of a multiproduct noncontinuous process with a single processing stage that has J non-identical, parallel, batch/semicontinuous units (j = 1, 2, …, J). The plant produces I distinct products (i = 1, 2, ..., I). The term "batch" is used to refer to both batch and semicontinuous units. For the latter, it is essentially a single campaign of one product. Other features of this process are:

1. A unit may process one or more products. Let $\boldsymbol{J}_i$ denote the set of units that may process product i, and let $\boldsymbol{I}_j$ denote the set of products that a unit j may process.
2. The batch size of a product may vary from batch to batch between some lower and upper limits that are product- and unit-dependent. Multiple units may process separate batches of the same product simultaneously. Two batches of a product, even if assigned to the same unit, need not follow each other.
3. The processing time of a batch varies linearly with batch size and its relationship depends on product-unit combination.
4. Switching from one batch to another may require transition time that is unit- and sequence-dependent. Because of quality or other incompatibilities, some job sequences may be forbidden on a unit.
5. Some units may not be available at the start of scheduling horizon, as they may need some time to process current batches. Similarly, it may not be possible to start processing some products at the start of horizon because of the lack of raw materials.

To accommodate these situations, unit-release times and job-release times are used as the earliest times at which a unit or product may start processing.

6. The operation is non-preemptive and there are no resource constraints.

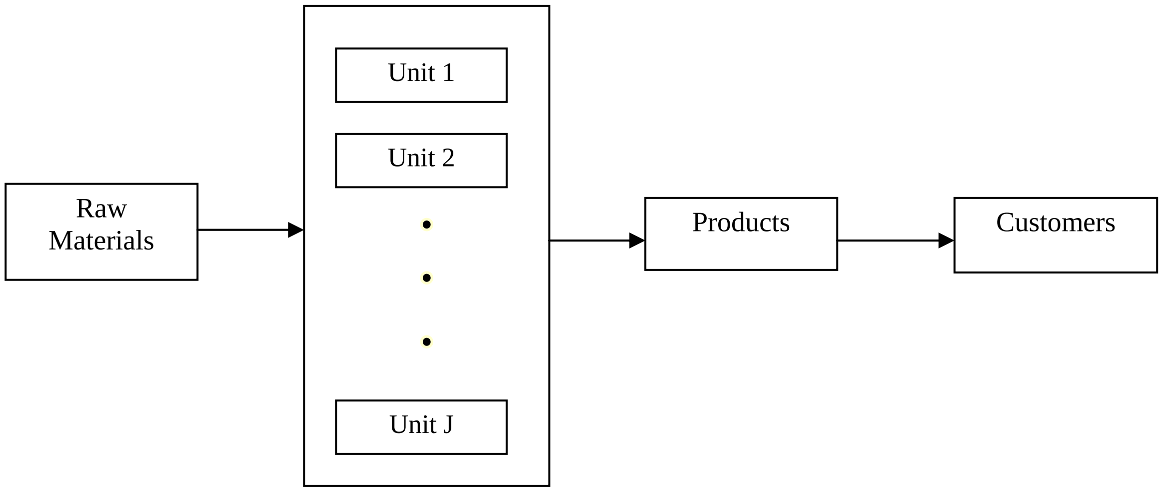


Figure 3.1: Schematic diagram of single stage multiproduct noncontinuous plant

While fulfilling the customer demands, two criteria are implemented. The first criterion is no due date in which there is no deadline on finished products and it is desired to finish the products as early as possible. In this case, makespan (the time at which the last product comes out of the processing stage) is the suitable objective. The second criterion is multiple due dates in which customers demand the products at different time points. If there is any failure in meeting the demands within the due dates, there will be some penalty on the delayed products and hence minimizing delay time is the appropriate objective in this case. Minimizing delayed amount is considered as another objective.

The plant has a total of O orders with D (d = 1, 2, …, D) due dates in the scheduling horizon. The due dates are sorted as $DD_1 < DD_2 < ... < DD_d$, where DD_d denotes the d^{th} due date, and let Q_{id} denote the total quantity of product i demanded at DD_d. An order may ask for multiple products at different due dates. A batch may fulfill multiple orders of a product. However, batches always fulfill orders in the order of their due dates. Similarly, multiple batches produced by different units may also fill a given order.

The scheduling problem involves (1) assignment of batches and their sequence on each unit, (2) determination of the start and end times of each batch and its size, and (3) allocation of batches to orders. Three objectives are chosen to test the models: (a) minimizing makespan (b) minimizing delay time (tardiness) (c) minimizing delayed amount.

In the next chapter, first, scheduling problem with no due date is addressed. Then, problem with multiple due dates ($D > 1$) is considered. For each of these problems, we consider processes with batch units and those with semicontinuous units. In other words, four MILP formulations are developed using slot-based and event-based approaches. Note that transition times are not considered while developing event-based models. And also in event-based approach, same product is treated as different tasks in different units. This means, product i that can take place in units 1 and 2, will be represented as task 1 and task 2 on unit 1 and unit 2 respectively. Otherwise, task is analogous to product in all other contexts. For further details about event-based representation, please refer to Ierapetritou et al. (1999).

Chapter 4

MODELS WITH SINGLE BATCH PER SLOT

4.1 Motivation

While developing short-term scheduling models using MILP, most researchers generally focus on the number of binary variables that are used in the model formulation. In most cases, the number of binary variables affects the computational effort of a MILP problem directly. To get a significant savings in binary variables, the following two approaches are used frequently: (1) representing binary variables more effectively (2) decomposing the existing binary variables. The first approach was widely used in scheduling formulations and the second one is being used in the past few years. For instance, Cerda et al. (1997) handled assignment and sequencing decisions through a tri-index binary variable (product-product-unit), while Mendez et al. (2000) decoupled the same binary variable into two different bi-index binary variables (product-product, product-unit) so that the number of binary variables was greatly reduced, though both the works use same approach (sequence-based method). But the same cannot be said for the work by Ierapetritou and Floudas (1998) and as well as the work by Maravelias and Grossmann (2003). They try to decouple task events from unit events in order to reduce the number of binary variables. Recently, Sundaramoorthy and Karimi (2004) have shown that such decoupling does not reduce the number of binary variables in a formulation. So in this chapter, the effect of decoupling on model formulation is also studied.

Besides decoupling, there is another issue that needs to be resolved. Among all continuous time representations, it is not very clear which approach is better than the others. Previously there were very few attempts to compare the existing continuous-time representations. Lim and Karimi (2003a) proved that slot-based formulations are better than sequence-based models. In this chapter, a comparison is drawn between the slot-based and the event-based models. Next, four simple slot-based mathematical formulations are developed. Then corresponding event-based models are formulated. Later, the slot-based models are compared with the event-based models.

4.2 MILP Formulations

A primary scheduling decision is to assign the production of various products to slots. Because production necessarily involves a unit, the decision naturally involves the assignment of unit as well. Two types of binary variables have been used in the literature for modeling these assignments. While most slot-based formulations (Lim and Karimi, 2003a; Lamba and Karimi, 2002a; Sundaramoorthy and Karimi, 2004) have directly used 3-index binary variables (product i on unit j during slot k), event-based formulations (Ierapetritou and Floudas, 1998) have decoupled these into two sets of 2-index binary variables (unit j at event n and product i at event n). Tri-index binary variables (Y_{ijk}) for slot-based formulations and two bi-index variables ($wv(t,n)$ and $yv(j,n)$) for event-based formulations are used in this chapter.

4.2.1 Semicontinuous Units & No Due Date

***Slot-based Formulation*:**

To assign products to slots, the following binary assignment variable is defined:

$$Y_{ijk} = \begin{cases} 1 & \text{if unit/line } j \text{ processes product } i \text{ during slot } k \\ 0 & \text{otherwise} \end{cases}$$

Exactly one product is assigned to each slot on each unit by using,

$$\sum_{i \in \boldsymbol{I}_j} Y_{ijk} = 1 \tag{4.1}$$

For modeling transitions (setups) between products in successive slots, a 0-1 continuous variable $X_{ii'jk}$ is used as follows.

$$X_{ii'jk} = \begin{cases} 1 & \text{if product } i' \text{ in slot } k \text{ and product } i \text{ in slot } (k-1) \text{ on unit/line } j \\ 0 & \text{otherwise} \end{cases}$$

There do exist other ways of modeling transitions, but Lim & Karimi (2004b) have shown that using the above transition variables results in tighter formulation. Clearly $X_{ii'jk} = Y_{ij(k-1)}Y_{i'jk}$ and this could be linearized exactly as,

$$Y_{ij(k-1)} = \sum_{i'} X_{ii'jk} \qquad k > 0 \qquad (4.2)$$

$$Y_{ijk} = \sum_{i'} X_{i'ijk} \qquad k > 0 \qquad (4.3)$$

Equations 4.2 and 4.3 ensure that only one product follows the other. Equation 4.2 ensures that if a product i is in slot (k–1) on unit j, then only one product follows i in slot k on that unit. Equation 4.3 ensures exactly the reverse. Note that no idle, zero or null products (unit is idle) are used in the slot-based model, so there are no transitions to/from the idle state.

Some assumptions are made while developing MILP formulation for scheduling semicontinuous processes. They are: (a) multiple products cannot be produced on a production line at the same time (b) all the units run at maximum production rates and known a priori (c) during transitions, no useful product is produced on any unit. With these assumptions, we explain the constraints involved in this model.

For product assignments and product transitions, equations 4.1 to 4.3 are used. It is assumed that products will be dispatched to customers only after finishing the processing of all the products as there are no intermediate due dates. This means that it does not matter how the campaign (the process of producing products for a certain amount of time) is run as long as the demands are met. Moreover, by running a campaign more than once, there will be an increase in the number of transitions and hence an increase in transition times/costs. So a product/campaign is not allowed to run more than once in a period.

Transition times are significant in practice because they disrupt the operation and hence reduce the profits of a plant. So to reduce the number of transitions and also to achieve desired yield or quality, units must run the campaigns for a certain minimum amount of time. This is called minimum production length of a campaign, and it is assumed that for every operation it is known a priori. And also the run length of a campaign, the time needed to finish that campaign, should exceed the minimum production length. When no campaign exists ($Y_{ijk} = 0$), the run length should be zero. The following constraints ensure the required.

$$RL_{ijk} \geq MPL_{ij}(Y_{ijk} - X_{iijs}) \qquad \forall k > 0, s = k, s > 1 \tag{4.4}$$

$$RL_{ijk} \leq H(Y_{ijk} - X_{iijs}) \qquad \forall k > 0, s = k, s > 1 \tag{4.5}$$

In the end, the total production of each product should meet the customer orders i.e. should meet or exceed the demand. Therefore,

$$D_i \leq \sum_{k} \sum_{j \in J_i} R_{ij}^U RL_{ijk} \tag{4.6}$$

If a product spans multiple slots, only the first slot is allowed to have the useful production and the remaining slots remain as null slots. So if a product i wants to continue in the next slot, then all the remaining slots are forced to have product i. The following constraint ensures the above. It also makes sure that all the null slots occur at the end of the horizon.

$$X_{iij(k+1)} \geq X_{iijk} \qquad \forall k > 1 \tag{4.7}$$

For no due date case, operation efficiency is the main criterion and hence minimizing makespan (*MS*) is considered as scheduling objective and it is defined as,

$$MS \geq \sum_{k} \left[\sum_{i} \sum_{i'} \tau_{ii'j} X_{ii'jk} + \sum_{i \in I_j} RL_{ijk} \right] \tag{4.8}$$

Event-based Formulation:

Binary variables in this representation are,

$$wv(t,n) = \begin{cases} 1 & \text{if task } t \text{ is processed in event point } n \\ 0 & \text{otherwise} \end{cases}$$

$$yv(j,n) = \begin{cases} 1 & \text{if unit/line } j \text{ is utilized in event point } n \\ 0 & \text{otherwise} \end{cases}$$

For assigning the products on each unit,

$$\sum_{t \in T_j} wv(t,n) = yv(j,n) \tag{4.9}$$

These constraints express that at each unit j and at an event point n only one of the tasks that can be performed on this unit should take place.

Sequence constraints are necessary because these equations provide the connections between starting and final times.

Same task in the same unit:

$$TS_{tj(n+1)} \geq TF_{tjn} - M(2 - wv(t,n) - yv(j,n)) \tag{4.10}$$

$$TS_{tj(n+1)} \geq TS_{tjn} \tag{4.11}$$

$$TF_{tj(n+1)} \geq TF_{tjn} \tag{4.12}$$

Equations 4.10 to 4.12 state that task t starting at event point ($n+1$) should start after the end of the same task performed in the same unit j which has already started at event point n.

Different tasks in the same unit:

$$TS_{tj(n+1)} \geq TF_{t'jn} - M(2 - wv(t',n) - yv(j,n)) \tag{4.13}$$

Eq. 4.13 states that task t if assigned to unit j in event point ($n+1$) starts only after the previous task t' finishes in event point n on unit j.

The limits on the duration of each event are given by,

$$TF_{tjn} - TS_{tjn} \leq Mwv(t,n) \tag{4.14}$$

$$TF_{tjn} - TS_{tjn} \geq MPL_{tj} wv(t,n) \tag{4.15}$$

Eq. 4.14 states that the duration of an event must be zero if it does not exist. If an event occurs, then Eq. 4.15 ensures that the duration of the event exceeds the minimum production length.

The total production of all the tasks related to the same product must meet the demand of that product. Therefore,

$$\sum_{t \in T_P} \sum_{j} \sum_{n} (TF_{tjn} - TS_{tjn}) R_{tj}^{U} \geq D_p \tag{4.16}$$

The objective is minimizing makespan and is given by,

$$MS \geq TF_{tjn} \qquad n = N \tag{4.17}$$

Having discussed the models for semicontinuous process using slot and event-based methods, we move on to the next scenario, batch units with no due date. First, MILP formulation is developed using the slot-based method and then the event-based formulation is discussed for batch units.

4.2.2 Batch Units & No Due Date

Slot-based Formulation:

For product assignments and transitions, eqs. 4.1 to 4.3 are used.

Because of the operational constraints and unit capacities, batch size of each product in each slot on each unit must be between some minimum and maximum limits and hence the following constraints are forced. Eq. 4.18 ensures that the amount produced of product i on unit j in slot k is not less than a certain amount (B_{ij}^{L}). If a product is assigned to multiple slots, then the production takes place in the first slot and the remaining slots will remain as null or idle slots. Eq. 4.19 ensures that the batch size does not exceed the maximum limit (B_{ij}^{U}).

$$B_{ijk} \geq B_{ij}^{L}(Y_{ijk} - X_{iijs}) \qquad \forall k > 0, s = k, s > 1 \qquad (4.18)$$

$$B_{ijk} \leq B_{ij}^{U}(Y_{ijk} - X_{iijs}) \qquad \forall k > 0, s = k, s > 1 \qquad (4.19)$$

To ensure that the total product produced on all units in all slots must meet the demand of that product,

$$D_i \leq \sum_{k} \sum_{j \in J_i} B_{ijk} \qquad (4.20)$$

The null slots are pushed to the end of the horizon using,

$$X_{iij(k+1)} \geq X_{iijk} \qquad \forall k > 1 \qquad (4.21)$$

The total length of a slot, if processing any useful product, consists of two parts. One is transition time and another is processing time. Processing time consists of FPT_{ij} that depends on the number of batches and VPT_{ij} that depends on the batch size. So the length of a useful slot k consists of FPT_{ij}, VPT_{ij} and transition time, if any. The scheduling objective is minimizing the makespan (*MS*) that is defined as the finishing time of final product and is given by,

$$MS \geq \sum_{k} \left[\sum_{i} \sum_{i'} \tau_{ii'j} X_{ii'jk} + \sum_{i \in I_j} (FPT_{ij} Y_{ijk} + VPT_{ij} B_{ijk}) \right] \tag{4.22}$$

***Event-based Formulation*:**

For assigning and sequencing, equations 4.9 to 4.13 are used. Besides these constraints, the following are needed.

Batch sizing decisions are taken by,

$$B_{tjn} \leq BS_{tj}^{U} wv(t,n) \tag{4.23}$$

$$B_{tjn} \geq BS_{tj}^{L} wv(t,n) \tag{4.24}$$

Equations 4.23 and 4.24 are analogous to eqs. 4.18 and 4.19 in slot-based formulation.

Starting time and ending time of each event point is given by,

$$TF_{tjn} = TS_{tjn} + FPT_{tj} wv(t,n) + VPT_{tj} B_{tjn} \tag{4.25}$$

The total production of all the tasks related to the same product must meet the demand of that product.

$$\sum_{t \in T_P} \sum_{j} \sum_{n} B_{tjn} \geq D_p \tag{4.26}$$

The scheduling objective is minimizing the makespan and is given by,

$$MS \geq TF_{tjn} \qquad n = N \tag{4.27}$$

With this the formulations are completed for short-term scheduling of both semicontinuous and batch processes with no due date using slot-based and event-based methods. Next, the necessary constraints that are to be added to the models presented are discussed to make them suitable for the case of multiple due dates.

4.2.3 Semicontinuous Units & Multiple Due Dates

The short-term scheduling of semicontinuous units with demands at multiple due dates is considered in this section. The time horizon for each unit comprises K slots, but the slots may not be identical on all units. These slots are designated to different due dates as follows. The first K_1 slots fill orders due at DD_1, but they need not finish before DD_1. Similarly, the first K_2 slots fill orders due on or before DD_2 but need not finish before DD_2, and likewise. So $K_1 \leq K_2 \leq K_3 \leq \ldots \leq K_d$. The same will be applied to event-based formulations.

Slot-based Formulation:

The run length of each campaign should exceed minimum campaign length and hence the following constraint is forced. Note that slots, though assigned to a due date, can extend beyond that due date. So unlike past works (Lamba and Karimi, 2002a; Lim & Karimi, 2003b) in which the run length of a campaign spans multiple slots if necessary, it needs a single slot for production in our model.

$$RL_{ijk} \geq MPL_{ij}(Y_{ijk} - X_{iijs}) \qquad \forall k > 0, s = k, s \notin FS \qquad (4.28)$$

With this, it is ensured that the production takes place only in the first slot if the product is assigned to multiple consecutive slots. And if there is no production in a slot, then it is made sure that the run length of that campaign is equal to zero in that slot. The following constraints ensure that.

$$RL_{ijk} \leq MY_{ijk} \qquad (4.29)$$

$$RL_{ijk} \leq M(Y_{ijk} - X_{iijs}) \qquad \forall k > 0, s = k, s \notin FS \qquad (4.30)$$

Note that unlike previous works (Karimi & McDonald, 1997; Lim & Karimi, 2003b), run length of a campaign is equal to slot length as a slot is allowed to go beyond a due date, if necessary.

The length of each useful slot is the sum of campaign run length and transition time, if any. TE_{jk} is the end time of slot k on line j and is given by,

$$TE_{jk} \geq TE_{j(k-1)} + \sum_{i} \sum_{i'} X_{i'ijk} CT_{i'ij} + \sum_{i \in I_j} RL_{ijk} \tag{4.31}$$

The total production on all lines should satisfy the demands that are due at the due date. Therefore,

$$\sum_{d'=1}^{d} Q_{id'} \leq \sum_{k=1}^{K_d} \sum_{j \in J_i} R_{ij}^{U} RL_{ijk} \tag{4.32}$$

Occasionally, demands cannot be satisfied on or before due date either because of high demand or short time. Then there will be delay in meeting the demands on or before the due date. Let D_{id} be the delay in filling the orders of product i due at demand window DD_d. If the last batch of product i in the first K_d slots finishes before DD_d, then D_{id} =0 (meeting the demands on time). If it does not, then D_{id} is the time by which the last batch is delayed beyond DD_d. Mathematically, this is expressed as,

$$D_{id} \geq TE_{jk} - DD_d - M(1 - Y_{ijk}) \qquad j \in J_i, k \leq K_d \tag{4.33}$$

In traditional parallel machine scheduling, researchers have mainly studied scheduling objectives such as makespan, flow time, tardiness/earliness, number of tardy jobs, etc. Two objectives are selected for this problem. The first one is minimizing the sum of weighted tardiness that is given by,

$$SWT = \sum_{d} \sum_{i} \alpha_{id} D_{id} \tag{4.34}$$

where α_{id} is the penalty for the delay in meeting the demand of product *i* at DD_d. This penalty may depend on Q_{id} and the importance of product *i*.

The second objective is minimizing the delayed amount. Delayed amount is the amount of products that could not be produced within the due date. The delayed amount is minimized after penalizing the delayed amount with a cost factor. Therefore,

$$SPCDA = \sum_{d}\sum_{i} C_i DA_{id} \tag{4.35}$$

SPCDA is the sum of product of cost factor (C_i) and delayed amount (DA_{id}). The second objective has chosen mainly to avoid big-*M* constraint that affects solution times.

Analogous to eq. 4.7, the following constraint is used to improve the tightness of the model.

$$X_{iij(k+1)} \geq X_{iijk} \qquad K_{(d-1)} + 1 \leq k < K_d \tag{4.36}$$

To calculate the second objective, we need to obtain the delayed amount of each product *i*. To calculate the delayed amount, first of all we have to ensure that every slot that belongs to a due date must end on or before the due date. Therefore,

$$TE_{jk} \leq DD_d \qquad k \leq K_d \tag{4.37}$$

Now, the delayed amount of a product *i* at DD_d is defined as,

$$DA_{id} \geq \sum_{d'=1}^{d} Q_{id'} - \sum_{k=1}^{K_d}\sum_{j \in J_i} R_{ij}^{U} RL_{ijk} \qquad j \in J_i, k \leq K_d \tag{4.38}$$

***Event-based Formulation*:**

For developing event-based model for the case of semicontinuous process with multiple due dates, the following are needed besides eqs. 4.9 to 4.15.

Eq. 4.16 is replaced with,

$$\sum_{t \in T_P} \sum_{j} \sum_{n=1}^{N_d} (TF_{tjn} - TS_{tjn}) R_{tj}^U \geq \sum_{d'=1}^{d} Q_{pd'} \qquad (4.40)$$

The delay if any, is given by,

$$D_{td} \geq TF_{tjn} - DD_d - M(1 - wv(t,n)) \qquad j \in J_t, n \leq N_d \qquad (4.41)$$

And the delayed amount is defined as,

$$DA_{pd} \geq \sum_{d'=1}^{d} Q_{pd'} - \sum_{t \in T_P} \sum_{j} \sum_{n=1}^{N_d} (TF_{tjn} - TS_{tjn}) R_{tj}^U \qquad (4.42)$$

And the scheduling objectives are given by eq. 4.34 and eq. 4.35.

4.2.4 Batch Units & Multiple Due Dates

Note that the models that are discussed in the section 4.2.2 can be modified from no due date case to multiple due dates case with the use of following constraints.

Slot-based Formulation:

In place of eqs. 4.18 and 4.19,

$$B_{ijk} \geq B_{ij}^L (Y_{ijk} - X_{iijs}) \qquad \forall k > 0, s = k, s \notin FS \qquad (4.43)$$

$$B_{ijk} \leq B_{ij}^U (Y_{ijk} - X_{iijs}) \qquad \forall k > 0, s = k, s \notin FS \qquad (4.44)$$

Eqs. 4.18 and 4.19 ensure that the production takes place only in the first slot if a product is assigned to multiple consecutive slots in a period.

In place of eq. 4.20,

$$\sum_{j \in J_i} \sum_{k=1}^{K_d} B_{ijk} \geq \sum_{d'=1}^{d} Q_{id'} \qquad (4.45)$$

And also eq. 4.7 is replaced with eq. 4.36.

The delayed amount is given by,

$$DA_{id} \geq \sum_{d'=1}^{d} Q_{id'} - \sum_{k=1}^{K_d} \sum_{j \in J_i} B_{ijk} \qquad j \in J_i, k \leq K_d \qquad (4.46)$$

Event-based Formulation:

Eq. 4.26 is replaced with the following.

$$\sum_{t \in T_P} \sum_{j \in J_t} \sum_{n=1}^{N_d} B_{tjn} \geq Q_{pd} \qquad (4.47)$$

And the scheduling objectives are given by eq. 4.34 and eq. 4.35.

Table 4.1 gives us the necessary constraints while developing the models for all scenarios.

Table 4.1: Model equations for scheduling noncontinuous chemical plants

Method	Case	Semicontinuous	Batch
Slot	N.D.D.	(4.1)-(4.8)	(4.1)-(4.3),(4.18)-(4.22)
	M.D.D.	(4.1)-(4.3),(4.28)-(4.38)	(4.1)-(4.3),(4.34)-(4.36),(4.43)-(4.46)
Event	N.D.D.	(4.9)-(4.17)	(4.9)-(4.13),(4.23)-(4.27)
	M.D.D.	(4.9)-(4.15),(4.34),(4.35), (4.40)-(4.42)	(4.9)-(4.13),(4.23)-(4.25), (4.34),(4.35),(4.47)

N.D.D = No Due Date, M.D.D. = Multiple Due Dates

4.2.5 Distribution of Slots to Due Dates

Past works generally allotted slots to due dates in two different ways. Some researchers (Lim & Karimi, 2003a) pre-assigned slots to due dates using some methodology/formula. But some others (Ierapetritou et al., 1999) considered iteration method for assigning slots to due dates. Both methods have advantages and disadvantages. By using iteration method, we can eliminate the unnecessary slots but at the same time need to run the model many times to fix the number of slots. If we use a fixed formula to find the number of slots, then there is always a chance of over estimating the number of slots. By increasing the number of slots, there will be an increase in number of binary variables and hence increase in CPU time.

In this work, the iterative procedure is used to find the number of slots. First, the number of slots is fixed with an initial guess and then the solution is found. Then the number of slots is increased by one and again the solution is obtained. The number of slots is increased until there is no change in the solution.

4.3 Slot-based vs. Event-based Models

Now, we solve several examples for each case (semicontinuous-no due date, multiple due dates; batch-no due date and multiple due dates) to evaluate the slot-based models and event-based models. We use GAMS 20.7/CPLEX 7.5.0 to solve all the models using Pentium 4 processor in this work. In all the examples, we assume that there is no transition times involved.

4.3.1 Example 1

One single stage semicontinuous plant processes 8 products with 3 parallel units. The data for minimum production length, maximum rate of production of each product on each unit are given in Table 4.2.

Table 4.2: Minimum production length and maximum production rate in Example 1

	Unit		
Product	J1	J2	J3
I1	10/20		
I2	12/20	15/25	
I3	12/25	10/20	
I4		15/20	10/20
I5			12/25
I6			15/20
I7		10/15	10/15
I8	10/15		

The demand for each product at the end of horizon is same (400). All units are empty and ready to process at the start of the horizon. We do not consider any setup times to keep the comparison simple. The scheduling problem involves finding the optimal sequence of products on each unit while minimizing the makespan.

Now, we solve this scheduling problem using both methodologies (Slot and Event). We use equal number of event points and slots (3). Table 4.3 shows the difference between slot-based model and event-based model.

Table 4.3: Model statistics for Example 1

Statistics	Event-based	Slot-based
Constraints	365	164
Continuous	73	145
Discrete	45	36
Optimum	58.6667	58.6667
Relaxation	26.67	54.4615
Nodes	6627	6
Iterations	210199	184
Non-zeros	1077	519
CPU time (s)	10.28	0.06

The number of binary variables is greater in event-based model and RMIP values are much better in slot-based model than in event-based model. In addition to RMIP values and binary variables, the other parameters such as CPU time, nodes, iterations, non-zeros and constraints also put slot-based model above event-based model.

4.3.2 Example 2

In this example, we consider one single stage batch plant that produces 8 products with 3 parallel units. The limits on batch sizes, processing times and demands for each product are given in Table 4.4 and Table 4.5 respectively. Unlike in Example 1, all units are pre-occupied with some product and the information is given in Table 4.5. Like in Example 1, there are no setup times.

Table 4.4: Batch size limits in Example 2

	Unit		
Product	J1	J2	J3
I1	50/100		
I2	100/140	100/150	
I3	100/150	100/150	
I4		80/120	100/150
I5			50/100
I6			100/200
I7		100/150	50/100
I8	100/200		

Table 4.5: Demands and Processing Times in Example 2

		Unit		
Product	Demand	J1	J2	J3
I1	400	0.15/3*		
I2	250	0.16/2	0.17/2	
I3	400	0.16/3	0.16/3*	
I4	500		0.15/2	0.14/3
I5	300			0.16/3
I6	250			0.17/2
I7	200		0.17/1	0.18/2*
I8	300	0.15/2		

Table 4.6: Model statistics for Example 2

Statistics	Event-based	Slot-based
Constraints	992	590
Continuous	328	577
Discrete	120	96
Optimum	-	158
Relaxation	72	151.59
Nodes	-	16984
Iterations	-	287382
Non-zeros	3096	2233
CPU time (s)	>10000	23.26

We have taken equal number of slots and event points (8) while solving the above scheduling problem using both models. The results are shown in Table 4.6 which again favors the slot-based models. Interestingly, the event-based model could not solve the problem in reasonable amount of time. Like in Example 1, RMIP value is much better in the slot-based model than the event-based model.

4.3.3 Example 3

This example is considered to address semicontinuous processes with demands at multiple due dates. Consider the semicontinuous plant in Example 1. In this example, the plant has to meet demands of various customers at different due dates ($D1$ =24 hr, $D2$ = 48 hr, $D3$ = 72 hr) as shown in Table 4.7. We consider that all products are equally important (α_{id} = 1).

We distribute the number of slots/event points among due dates in the following way to solve this scheduling problem using slot-based and event-based methods: K_1 = 3, K_2 = 6, K_3 = 9.

Table 4.7: Demands of various products in Example 3

	Demand		
Product	D1	D2	D3
I1	100	100	100
I2	100	100	100
I3	100	200	100
I4	200	100	200
I5	200	300	100
I6	150	200	200
I7	100	100	100
I8	200	200	100

As in the previous example, the event-based model could not solve this scheduling problem but the slot-based model could solve the above problem in about

637s CPU time. RMIP value is zero for both models. As was the case with previous examples, the slot-based model has fewer binary variables than the event-based model. The results are shown in Table 4.8.

Table 4.8: Model statistics for Example 3

Statistics	Event-based	Slot-based
Constraints	1603	961
Continuous	325	703
Discrete	135	108
Optimum	-	32.6667
Relaxation	0	0
Nodes	-	221315
Iterations	-	3959274
Non-zeros	4975	3286
CPU time (s)	>10000	632.7

4.3.4 Example 4

In this example, we consider batch processes with products having demands at multiple due dates. A single stage batch plant processes 8 products with 3 units in parallel. Three due dates are considered ($D1$ =50 hr, $D2$ = 100 hr, $D3$ = 200 hr). All products have different demands at all due dates and the details are shown in Table 4.9. The same data is used that was used in Example 2 for processing times and batch size limits. We have distributed 9 slots/event points to due dates in the following manner: $K_1 = 3$, $K_2 = 6$ and $K_3 = 9$.

Table 4.10 compares the performance of both models. The slot-based model able to solve the model but the event-based model again failed to finish. The objective value achieved by slot-based model is 23 units. RMIP value is same for both models and it is zero. Again slot-based model has fewer binary variables than event-based model. Overall we observed that slot-based model performs better than event-based model.

Table 4.9: Demands of various products in Example 4

	Demand		
Product	D1	D2	D3
I1	100	100	100
I2	100	100	100
I3	100	200	100
I4	50	100	100
I5	100	100	100
I6	100	200	100
I7	100	100	100
I8	50	100	100

Table 4.10: Model statistics for Example 4

Statistics	Event-based	Slot-based
Constraints	1279	853
Continuous	424	703
Discrete	135	108
Optimum	-	23
Relaxation	0	0
Nodes	-	271845
Iterations	-	7977923
Non-zeros	4111	3178
CPU time (s)	>10000	1351.11

4.4 Discussion

In event-based method, which uses the concept of decoupling, the same product on two machines is considered as two different tasks. So the binary variables result from *wv(i,n)* are *IJN* and hence in reality the total number of binary variables that result from *wv(i,n)* and *yv(j,n)* are (*IJN*+*JN*) though it looks like (*IN*+*JN*) variables. In slot-based method the total binary variables result from *Y(i,j,k)* are *IJK*. If we consider the same number of event points (*N*) and slots (*K*), then slot-based model results in fewer binary variables than event-based model. The difference between decoupling and not decoupling is that the later displays the unit information explicitly in terms of *j*, while the former hides the same behind *i*. By employing the concept of decoupling, there is an increase of *JN* number of binary variables in event-based models.

Some past works such as Ierapetritou et al. (1999), while comparing different models, did not consider computational issues such as software version and the performance of hardware. They compared two different models on two different hardware and software versions – this makes the comparison unfair. It is obvious that new hardware works better than the old one and it is very clear that we should not compare different models on different hardware. Keeping this in view, we have compared both models using same version of GAMS/CPLEX and using same PC to avoid the ambiguities regarding computational issues.

4.5 Conclusions

In this chapter, four simple slot-based mathematical models are presented for short-term scheduling of batch and semicontinuous processes with no due date and with multiple due dates. We have developed corresponding event-based models using the concept of decoupling. Later, slot-based models are compared with event-based models and we observed that the results put slot-based models far ahead of event-based models. Especially RMIP values are very much in favor of slot-based models. So we conclude that slot-based models are better than their correspondent event-based models. And also we observed that decoupling of tasks from units in a mathematical formulation cannot reduce the number of binary assignment variables.

But so far, in the literature and in this chapter, batches are treated individually while scheduling the batch processes. This approach results in large number of binary variables for scheduling large number of batches waiting before the units. So in the next chapter, we address a novel way of scheduling single-stage multiproduct batch plants with demands at multiple due dates to solve larger size problems.

CHAPTER 5

BATCH UNITS - MODELS WITH MULTIPLE BATCHES IN A SLOT

5.1 Motivation

The semiconductor manufacturing industry is one of the biggest as well as one of the most complex industries. It is one of the major industries in Singapore. Semiconductor manufacturing in general, and wafer fab operations in particular, present challenging scheduling and control problems. The characteristics that make the semiconductor manufacturing difficult to schedule are as follows: reentrant flows, unpredictable yield and rework time at critical operations, batching, shared resources, high mix of products and rapidly changing technologies that may change the path of the product.

The entire semiconductor manufacturing process consists of five major steps. The first step is wafer manufacturing in which high purity silicon wafer slices are made. First, a highly pure silicon crystal is grown in a crystal growth chamber. Then it is sliced into thin wafers. After this step, they enter into wafer fabrication unit.

The wafer fabrication process, wafer fab in short, dominates the economics of IC production and it is the most technologically-complex and capital-intensive stage in semiconductor manufacturing. Polished wafers are fed into this section. Electronic circuits are developed on a polished and clean wafer in a clean room. This is done in multiple passes. In each pass, a new layer with a pattern is formed on the wafer. Each pass in a wafer fabrication process involves some or all of the following operations: (a) Deposition (b) Lithography (c) Etching (d) Resist strip (e) Ion implant.

In deposition step, SiO_2 layer is deposited by means of chemical vapor deposition in a batch furnace. Lithography is a process used to create multiple layers of circuit patterns on a chip. First the wafer is coated with a light sensitive chemical called photoresist in a spinner coater assembly. The wafer is exposed to the UV light through the reticle that contains the pattern for a few chips. The photoresist changes its composition where it is exposed to the beam. This entire process is carried out in a stepper.

In order to define the circuits, the exposed material is etched away in a solvent filled developer bath. After the etching step, selected impurities are introduced in a controlled fashion to change the electrical properties of the exposed portion of the layer. This is called ion implantation. The remaining photoresist on the wafer is removed by a process similar to etching. The individual circuits are tested electrically by means of thin probes. Circuits that fail to meet specifications are marked with an ink dot. Wafers are then cut into individual circuits, which are also called dies. Dies are wide bounded by golden strings and moulded with land frame in plastic or ceramic packages that protect them from environment. Chips must be defect-free. So they undergo extensive testing in a testing unit to ensure that they are defect-free before coming to the market.

In the wafer fab, the wafers are processed in terms of lots (a lot contains a fixed number of wafers) in order to build up layers of patterns to produce the required circuitry. As discussed, this involves a complex sequence of processing steps with a number of operations that require different kinds of equipment (Lamba and Karimi, 2000; Kim et al., 2002). One such step is the photolithography.

Photolithography is the most complex operation in wafer fabrication, and it requires the greatest precision. In this process, the wafer is exposed to a light source that passes through the reticle (or mask) that holds the pattern of the circuitry for a particular layer. The main steps that a wafer has to undergo during a photolithography step are coating of the photo resist, exposure of the resist and the development of the resist. In the first step, the wafer is spun while the resist is deposited onto the wafer. The wafer is then baked to firm the photoresist and improve its adhesion to the wafer. The wafer is exposed to the UV light through the reticle that contains the pattern for a few chips. This alignment and exposure step is repeated until the whole wafer surface is exposed. Simply in this process, the wafer is exposed to a light source that passes through the reticle (or mask) that holds the pattern of the circuitry for a particular layer.

The steppers used in the photolithography process are very expensive machines and are usually the bottleneck in the manufacturing process. One-third of the total work-in-process (WIP) competes to get processed at the steppers and hence it is very critical to allocate the wafer lots to the steppers in an efficient manner. To fabricate an integrated chip (IC) from a silicon wafer, one needs to build several layers of circuitry. For each layer of a wafer, one photo operation is required which is carried out in a stepper machine. For each stepper machine there may be several types of lots waiting in the buffer for different layers to be processed. Often steppers are shared among different types of wafer lots for various layers. Large number of lots belonging to similar device types waits in the buffer for hours to be processed on the stepper machines, which poses serious scheduling problems. Since the stepper machines regulate the total throughput of the fab, it is very crucial to schedule the large number of lots waiting in the photolithography process with an efficient and powerful mathematical model.

Past scheduling work on such a process addressed the problem in terms of individual batches. The problem with such an approach is that it becomes impossible to schedule large number of lots waiting before the steppers. For example consider a wafer fab that manufactures two devices/products consisting of 4 layers each. For each layer, there are a number of lots (batches in thousands) waiting before the steppers which are needed to be scheduled for maximum throughput. Previous works (Lim & Karimi, 2003a; Mendez et al., 2000) could not solve the above problem in reasonable amount of time. Thus, there is a need to develop a mathematical model that addresses the problem in terms of batches of integral lots suitable for similar machines.

5.2 Model Formulation

The main features of the two proposed models described below are (a) the use of continuous variable to allow transitions on production units (b) handling of multiple batches in a single slot. The way in which the batches are accommodated which belong to the same product, makes both models different from each other. As will be shown, both models can easily account for changeovers and can handle many batches of different products.

For both formulations, the time horizon for each unit comprises K slots, but the slots may not be identical on all units. These slots are designated to different due dates as follows. The first K_1 slots fill orders due at DD_1, but they need not finish before DD_1. Similarly, the first K_2 slots fill orders due on or before DD_2 but need not finish before DD_2, and likewise. So $K_1 \leq K_2 \leq K_3 \leq \ldots \leq K_d$. Using an iterative procedure, the number of slots to each due date are fixed. Now, various constraints are discussed in both models, where, unless otherwise stated, all constraints are to be written for all values of their indices.

5.2.1 Model M1

Assignments: To assign products to slots on each unit, one binary variable Y_{ijk} is defined in the following way.

$$Y_{ijk} = \begin{cases} 1 & \text{if unit } j \text{ processes product } i \text{ during slot } k \\ 0 & \text{otherwise} \end{cases}$$

To allow only one product in each slot on each unit,

$$\sum_i Y_{ijk} = 1 \tag{5.1}$$

Eq. 5.1 assigns exactly one product on each unit in each slot. By doing this, every slot will have one useful product on each unit. Note that there is no idle/dummy product in our model formulation. Even though there is a product in each slot on all the units, there is no useful production in some slots. These slots are called null slots. This means that the products are allowed to span multiple slots but the null slots are ensured to have no useful production. We will discuss about which slots are null and which slots are useful i.e. slots that produce products to meet the demands and also how we force the products to span multiple slots, if necessary.

Transitions: To allow one product to follow another, we use one continuous variable that always takes binary values in the following way.

$$X_{ii'jk} = \begin{cases} 1 & \text{if product } i' \text{ follows product } i \text{ in slot } k \text{ on unit } j \\ 0 & \text{otherwise} \end{cases}$$

As shown earlier, this can be linearized exactly as,

$$Y_{ij(k-1)} = \sum_{i'} X_{ii'jk} \qquad \forall k > 0 \qquad (5.2)$$

$$Y_{ijk} = \sum_{i'} X_{i'ijk} \qquad \forall k > 0 \qquad (5.3)$$

Slot length: Let N_{ijk} and B_{ijk} be the number of batches and the amount of product *i* on unit *j* in slot *k* respectively. The length of each useful slot is defined as the sum of processing time (fixed and variable) and transition time, if any. Let TE_{jk} denote the time at which slot *k* ends on unit *j*. Then TE_{jk} is defined as,

$$TE_{jk} \geq TE_{j(k-1)} + \sum_{i}\sum_{i'} X_{i'ijk} CT_{i'ij} + \sum_{i} (N_{ijk} FPT_{ij} + B_{ijk} VPT_{ij}) \qquad (5.4)$$

$(N_{ijk}FPT_{ij} + B_{ijk}VPT_{ij})$ is the total processing time of B_{ijk}, where FPT_{ij} and VPT_{ij} are parameters. FPT_{ij} depend on the number of batches and VPT_{ij} depend on the batch size. So the length of a useful slot *k* consists of fixed processing time, variable processing time and transition time if any.

Boundaries on N_{ijk} and B_{ijk}: Often, because of operational constraints we may not process more than a fixed number of batches and hence we impose the following boundary constraints.

$$N_{ijk} \leq N_{ij}^{U} Y_{ijk} \qquad \forall k > 0, j \in J_i \qquad (5.5)$$

$$N_{ijk} \leq N_{ij}^{U} (Y_{ijk} - X_{iijs}) \qquad \forall k > 0, s = k, s \notin FS \qquad (5.6)$$

Eq. 5.6 ensures that the first slot among the allotted slots will have useful production if assigned to a product and remaining slots will be null slots.

We force the following limits on the amount to be produced in each slot.

$$B_{ijk} \leq B_{ij}^{U} N_{ijk} \qquad \forall k > 0 \qquad (5.7)$$

$$B_{ijk} \geq B_{ij}^{L} N_{ijk} \qquad \forall k > 0 \qquad (5.8)$$

Note that B_{ijk} is zero in null slots as the number of batches is forced to be zero according to eq. 5.6.

Demand: The amount produced should satisfy the orders due at DD_d and earlier. So we should force the following constraint to satisfy the demands at the end of each period respectively.

$$\sum_{k=1}^{K_d}\sum_{j\in J_i} B_{ijk} \geq \sum_{d'=1}^{d} Q_{id'} \qquad j\in J_i, k \leq K_d \qquad (5.9)$$

We calculate the total amount of each product produced before a due date DD_d by summing the amount of *i* produced over all the slots and all the units before that due date. Eq. 5.9 ensures that the amount produced will meet the demand Q_{id} but not necessarily on or before due date *d*.

Delay time: Let D_{id} be the delay in filling the orders of product *i* due at demand window DD_d. Mathematically, we express this as,

$$D_{id} \geq TE_{jk} - DD_d - M(1-Y_{ijk}) \qquad j\in J_i, k \leq K_d \qquad (5.10)$$

Initial Plant State: At the start of the scheduling horizon, units may be processing some batches. New products assigned to a unit cannot start until the current batch, if any, on that unit finishes (non-preemptive operation). To ensure this, we set

$$TE_{j0} = RT_j \qquad (5.11)$$

where RT_j, called the unit release time, is the time at which the current batch of product *i* finishes on unit *j* or the time at which the unit is ready to process the required product.

Scheduling Objectives: We select two objectives for this scheduling problem. The first one is minimizing the sum of weighted tardiness and the second one is minimizing delayed amount that are discussed in chapter 4 and are given by eq. 4.34 and eq. 4.35 respectively.

So we need equations 5.1 to 5.11, 4.34 and 4.36 to calculate the first objective i.e. minimizing the weighted sum of tardiness and equations 5.1 to 5.8, 5.11, 4.35 to 4.37 and 4.46 for minimizing the delayed amount in Model M1.

5.2.2 Model M2

Model M2 differs from Model M1 in representing the number of batches and batch size. So we discuss the constraints regarding the number of batches, batch size and also timing constraints which need to be redefined. All other constraints are same as in Model M1.

Number of Batches and Batch Size: We allow only first slot among the slots that are present in a production period to have useful production and make all the remaining slots null. We allow the useful slot to take more than one batch of the same product. But in model M1, all slots may have zero number of batches.

The number of batches of product i on unit j in slot k is defined as,

$$N_{ijk} = N_{ij}^{L}(Y_{ijk} - X_{iijs}) + \Delta N_{ijk} \qquad \forall k > 0, s = k, s \notin FS \qquad (5.12)$$

$$\Delta N_{ijk} \le (N_{ij}^{U} - N_{ij}^{L})(Y_{ijk} - X_{iijs}) \qquad \forall k > 0, s = k, s \notin FS \qquad (5.13)$$

N_{ij}^{L} and N_{ij}^{U} are the lower and upper bounds on the number of batches respectively and they are both unit- and product-dependent. By defining the number of batches in the above way, we ensure that only the first slot among the slots assigned to a product has useful production if that product spans multiple slots in a period. For doing so, $X_{iijs} = 1$ if the slot is not the first one. So, $N_{ijk} = N_{ij}^{L}Y_{ijk} + \Delta N_{ijk}$ for slots that are first in their production schedule because $X_{iijs} = 0$ for $s = 1$ and for all remaining slots in that product campaign, X_{iijs} = 1 and hence the number of batches in those slots for that product i is zero.

In the same way we define the amount to be produced in the useful slots. Let B_{ijk} be the total amount of product i on machine j in slot k. We define B_{ijk} as,

$$B_{ijk} = [N_i^L(Y_{ijk} - X_{iijs}) + \Delta N_{ijk}]B_{ij}^L + \Delta B_{ijk} \qquad \forall k > 0, s = k, s \notin FS \qquad (5.14)$$

$$\Delta B_{ijk} \le [N_i^L(Y_{ijk} - X_{iijs}) + \Delta N_{ijk}](B_{ij}^U - B_{ij}^L) \qquad \forall k > 0, s = k, s \notin FS \qquad (5.15)$$

B_{ij}^U and B_{ij}^L are the lower and upper bounds on batch size respectively and both are unit- and product-dependent. So the amount of a batch that belongs to product i produced on a unit j in a slot k is equal to ($B_{ij}^L + \Delta B_{ijk}$). Like the number of batches, the amount of product is also zero in those slots that are not the first in their production campaign.

Slot length: As there is a change in the way we represent the number of batches in this formulation, we modify timing constraints accordingly. Having defined the number of batches and the amount to be produced in each slot, we define the length of each slot in model M2. Let TE_{jk} denotes the time at which slot k ends on unit j. Then, TE_{jk} is defined as,

$$\begin{aligned} TE_{jk} \ge TE_{j(k-1)} &+ \sum_i \sum_{i'} X_{i'ijk} CT_{i'ij} + \sum_i (N_i^L(Y_{ijk} - X_{iijs}) + \Delta N_{ijk}) FPT_{ij} \\ &+ \sum_i ((N_i^L(Y_{ijk} - X_{iijs}) + \Delta N_{ijk}) B_{ij}^L + \Delta B_{ijk}) VPT_{ij} \end{aligned}$$

$$\forall k > 0, s = k, s \notin FS \qquad (5.16)$$

Note that we defined B_{ijk} and N_{ijk} variables for better understanding of our model M2. While building the model M2, we do not use these variables and we replaced these two variables using equations 5.18 and 5.20.

Demand: Demand constraint also differs from model M1 as there is a change in representing the amount of product to be produced.

$$\sum_{k=1}^{K_d}\sum_{j\in J_i}((N_i^L(Y_{ijk}-X_{iijs})+\Delta N_{ijk})B_{ij}^L+\Delta B_{ijk})\geq\sum_{d'=1}^{d}Q_{id'} \qquad k\leq K_d, s\notin FS, s=k \qquad (5.17)$$

Miscellaneous: The delayed amount is defined as,

$$DA_{id}\geq\sum_{d'=1}^{d}Q_{id'}-\sum_{k=1}^{K_d}\sum_{j\in J_i}((N_{ij}^L(Y_{ijk}-X_{iijs})+\Delta N_{ijk})B_{ij}^L+\Delta B_{ijk})$$

$$k\leq K_d, s\notin FS, s=k \qquad (5.18)$$

So we need equations 5.1 to 5.3, 5.10, 5.11, 5.13, 5.15 to 5.17, 4.34 and 4.36 for tardiness as objective and equations 5.1 to 5.3, 5.11, 5.13, 5.15, 5.16, 5.18 and 4.35 to 4.37 for delayed amount as objective while building the model M2.

5.3 Model Evaluation

We illustrate the effectiveness of models M1 and M2 with four working examples. These examples with different problem sizes and complexities will demonstrate the capabilities and efficacy of our methodology. We have taken two examples directly from the literature and added two more examples to show the vast application of our approach. We compare all our examples with Lim and Karimi (2003a) and did not compare with other major works such as Mendez et al. (2000) and Cerda et al. (1997) because Lim and Karimi (2003a) proved that their model performs better than Mendez et al. (2000) and Cerda et al. (1997). And also Lim and Karimi (2003a) concluded that their general model i.e. multiple orders per product model (we address this model as L&K model from now on) is better than the special case of single order per product as the former gives a better optimal solution than the latter. So we compare our models with their general model in all our examples. The models are formulated in GAMS 20.7 and solved by CPLEX 7.5.0 on a 2.40 GHz Pentium 4 PC.

5.3.1 Example 1

We have taken this example from Mendez et al. (2000). A multiproduct plant has seven units (J=7) and produces eight products (I=8). In this problem, the plant has to deliver products at seven due dates (D=7). For full details of this example, please refer to the work by Mendez et al. (2000). The scheduling objective is to minimize tardiness.

Mendez et al. (2000) use a two-step procedure to solve this problem. The first step makes the batching decisions to minimize the inventory. However, in this step, they require all orders to complete before their due dates. Their second step schedules the batches obtained from the first step to minimize the tardiness. L&K model solved this example in a single step using their general model. They also solved this example using their special case of one order per product model using the batches obtained by Mendez et al. Now, we solve this example using model M1 and model M2.

In this example, the batch size is constant and the batch size is more than demands for some products at some due dates. So in this case, we really do not need to assign slots for every due date for each unit. We have taken the number of slots equal to 3 and assigned to each due date as K_1=1, K_2=2, K_3=3, K_4=3, K_5=3, K_6=3, K_7=3.

In L&K model, they discussed about big-M constraints and the effect of Big-M on the model. We solve our both models with big-M and the results obtained shows that our model is very robust with respect to big-M value. We compared our model with L&K model and the results presented in Table 5.1 show the effectiveness of our both models. Both models solved the problem at hand easily as did L&K model. But M2 appeared to solve the problem in very efficient manner than M1 and L&K models. Model statistics that are shown in Table 5.2 reflects the performance of each model.

Both M1&M2 have the same number of binary and continuous variables. M2 has fewer constraints than M1 and L&K. Even though discrete variables are more in models we proposed compared to L&K model, binary variables are fewer than L&K model. It seems that integer variables that are part of discrete variables in our models do not have much effect on the performance of a model. However our model has more non-zeros and continuous variables that do not have much effect on model performance compared to binary variables.

Table 5.1: Big-*M* effect on models M1, M2 and L&K in Example 1

M	Nodes			Iterations			CPU time(s)		
	M1	M2	L&K	M1	M2	L&K	M1	M2	L&K
200	16	16	60	495	566	2131	0.56	0.53	2.28
500	78	10	34	1597	496	1371	1.39	0.64	1.78
1000	20	48	29	606	1019	1071	0.67	0.98	1.62
2000	116	44	34	2032	1094	1369	1.77	1.11	1.84
5000	61	57	55	1302	1205	2158	1.16	1.19	2.2
10000	72	32	47	1407	956	1815	1.37	0.99	2.18
100000	98	49	47	2180	984	1815	1.72	1.05	2

Table 5.2: Model statistics for Example 1

Model	Discrete variables	Continuous variables	Constraints	Non-zeros
M1	282	1284	1640	7662
M2	282	1284	1358	7743
L&K	223	327	1769	8619

Table 5.3 summarizes the optimal schedules obtained from all three models. Tardiness value, which is the scheduling objective, is zero for all three models in both RMIP and MILP solutions. However, they differ in the allocation of batches to units.

Overall, we place M2 ahead of M1 and M1 ahead of L&K with respect to model statistics and *M*-value.

Table 5.3: Optimal schedules for Example 1

	M1				M2				L&K			
unit	product/ batches	batch size (kg)	start-finish (h)	due date (h)	product/ batches	batch size (kg)	start-finish (h)	due date (h)	product	batch size (kg)	start-finish (h)	due date (h)
1	7/1	6000	0.0-13.1	24	2/1	6000	0.0-14.8	72	3	6000	0-13.4	48
					1/1	6000	14.8-24.3	72	4	6000	20.9-36	72
					4/2	6000	24.3-49.3	72	1	6000	36-46.1	72
2	5/1	6000	0-11.1	24	2/1	6000	0-10	48	2	6000	0-10	48
					1/1	6000	18.0-27.5	72	6	6000	10-30.1	48
					1/1	6000	27.5-35.5	72	6	6000	30.1-46.1	120
									1	6000	46.1-58.1	72
3	8/1	6000	0.0-21.6	24	7/1	6000	0.0-15.2	24	5	6000	7.3-19.5	24
	4/3	6000	21.6-60.4	72	6/1	6000	15.2-35.3	48	7	6000	19.5-33.6	120
					8/1	6000	64.6-86.4	144	3	6000	33.6-47	48
4	3/2	12000	0.0-25.6	48	3/2	12000	8.0-33.6	48	1	6000	0-8	72
	7/1	6000	25.6-41.0	72	7/1	6000	45.6-61	168	7	6000	16-31.2	96
									6	6000	31.2-51.3	168
5	6/2	5000	56.6-90.0	96	5/1	5000	0-8	24	4	5000	8-25.9	48
					7/2	4500	8.0-34.1	72	4	5000	25.9-37.9	48
					6/1	5000	44.4-64.5	168				
6	6/1	5000	0.0-20.1	48	4/1	5000	0.0-13.7	48	7	4500	0-12	24
	4/1	5000	20.1-33.5	48	6/1	5000	13.7-30.7	120	7	4500	12-24	120
	7/2	9000	33.5-58.6	120	5/1	5000	55.8-65.5	72	4	5000	24-37.7	120
									4	5000	37.7-49.7	168
7	8/1	6000	0.0-21.5	48	8/1	6000	0.0-21.5	48	8	6000	0-21.5	24
	2/2	6000	21.5-42.7	48,72	4/1	6000	50.5-64.1	120	2	6000	38.1-49.3	72
	1/3	6000	42.7-68.2	72					8	6000	49.3-70.9	96

5.3.2 Example 2

In Example 1, it is relatively easy to meet all due dates and hence it is difficult to identify a better model that needs less computation effort. So we consider another example where it is not possible to meet all the demands within the due dates. This example is taken from Lim and Karimi (2003a). It involves four products, three units and four due dates. We consider initial plant state in this example and hence the solution differs from their work. Both tardiness and delayed amount are considered as scheduling objectives in this example.

The batch size is not constant and varies between minimum and maximum limits. We have taken the minimum and maximum limit on number of batches as 1 and 5 respectively in this example for model M2 and maximum limit on number of batches is 5 in model M1. For further details such as processing times, transition times and demands at various due dates for different products, please refer to the work by Lim and Karimi (2003a). We have taken K_1=1, K_2=2, K_3=3, K_4=5 to solve this problem.

We solve this example for a wide range of *M*-values (Table 5.4) and the schedules obtained are shown in Table 5.6. L&K model is very much erratic computationally and largely dependent on *M*-value where as our models M1 and M2 have shown tremendous robustness with respect to *M*-value and it is evident from the results. But still there is no evidence how one model's performance depends on *M*-value. Table 5.5 draws a comparison among all models (M1, M2 and L&K) for both objectives. Computationally both objectives put M1 above all even though there is not much difference between M1 and M2 but L&K model clearly stands far behind in both cases.

Table 5.4: Big-*M* effect on model performance in Example 2

M	Nodes			Iterations			CPU time (s)		
	M1	M2	L&K	M1	M2	L&K	M1	M2	L&K
100	630	1064	28864	15680	17315	581677	1.60	1.73	65.51
500	900	988	34176	13381	16690	852883	1.61	1.53	76.48
1000	658	950	32727	13960	15983	645370	1.40	1.64	92.76
2000	1003	2309	157749	20452	38123	4089090	2.01	3.13	440.9
5000	1575	784	159943	35048	14194	4761510	3.10	1.48	499.9
10000	865	2782	426651	16910	53560	15144934	1.90	4.15	1610

For solving this example with tardiness as scheduling objective, both of our models have taken less computational time than L&K model. Our models, in spite of having integer variables proved to be computationally more effective mainly due to the reduction of binary variables. We can attribute the reduction in binary variables mainly to two factors. One factor is using fewer time slots and the other one is accommodating multiple batches in single slot. The objective value as well as the optimal schedule (Table 5.6) is same for all three models.

Table 5.5: Model statistics for Example 2

	tardiness (for M=100)			delayed amount		
Statistics	M1	M2	L&K	M1	M2	L&K
Constraints	528	360	531	680	328	513
Continuous	209	209	100	209	209	100
Binary	40	40	77	40	40	77
Integer	40	40	0	40	40	0
Optimum	33.87	33.87	33.87	17517	17517	17517
Relaxation	0	0	0	0	0	0
Non-zeros	1842	1432	1766	1596	1186	1474
Nodes	630	1064	28864	1135	2208	50192
Iterations	15680	17315	581645	20283	24678	1682287
CPU time (s)	1.60	1.73	65.51	1.68	2.14	96.12

Table 5.6: Optimal schedules for Example 2

	M1				M2				L&K			
unit	product/ batches	batch size (kg)	start-finish (h)	due date (h)	product/ batches	batch size (kg)	start-finish (h)	due date (h)	product	batch size (kg)	start-finish (h)	due date (h)
1	3/1	150	0.0-28.9	24,48	3/1	150	0.0-28.9	24,48	3	150	0.0-28.9	24,48
	2/1	102.25	28.9-48.53	48	2/1	102.25	28.9-48.53	48	2	102.25	28.9-48.53	48
	3/1	100	48.53-69.83	72	3/1	100	48.53-69.83	72	3	100	48.53-69.83	72
	2/2	100	69.83-107.13	96	2/2	100	69.83-107.13	96	2	100	69.83-89.13	96
									2	100	89.13-107.13	96
2	1/1	100	0-20	24	1/1	100	0.0-20	24	1	100	0.0-20	24
	4/2	100	20-58	48	4/2	100	20-58	48	4	100	20-39.5	48
	4/1	100	58-76.5	72	4/1	100	58-76.5	72	4	100	39.5-58	48
	4/1	100	76.5-95	96	4/1	100	76.5-95	96	4	100	58-76.5	72
									4	100	76.5-95	96
3	2/1	101.62	0-19.75	24	2/1	101.62	0.0-19.75	24	2	101.62	0-19.75	24
	1/1	150	19.75-48	48	1/1	150	19.75-48	48	1	150	19.75-48	48
	2/1	96.13	48-72	72	2/1	96.13	48-72	72	2	96.13	48-72	72
	1/1	140	72-98.8	96	1/1	140	72-98.8	96	1	140	72-98.8	96

With minimizing delayed amount as objective, the gap between computational time of our models and L&K model further increased even though the objective value is again same for all three models. Note that the formulation is free of big-*M* and hence we did not solve the problem for different values of *M*. As the models are free of big-*M*, this objective may be the accurate criterion to compare all three models. We can solve this example with minimizing delayed amount as objective with 4 slots. But to keep the comparison fair using both the objectives we compared our models with five slots and for L&K model, we used their own formula and kept the number of slots at 7. From Table 5.4 and Table 5.5, we observed that model M1 has an edge over model M2 and both of our models have performed very well than L&K model in this example.

5.3.3 Example 3

This example is the motivating example behind this work and has taken mainly to show the efficacy of our models in solving bigger size problems where a large number of batches need to be processed on parallel machines. We assumed the following so that we apply our formulation to apply our methodology to schedule the lots in photolithography area in Semiconductor industry.

1. Lots that belong to different layers are available in sufficient quantity to process using stepper machines for a given time.
2. We consider lots belonging to the same layer as batches belong to same product. So each product represents each layer. If any lot that has processed for a layer returns to stepper machines then we consider it as a lot that belongs to the next product. Implicitly, we did not consider precedence relationships between adjacent layers.

3. We did not consider resource constraints. So we assume that masks are always available to units to process lots in sufficient quantity.
4. We consider all lots that belong to different layers of devices as the batches belong to different products. For example, two different devices that have four layers each are considered as eight different products having different demands at different due dates in our work.

With these assumptions, we now proceed to demonstrate the effectiveness of our models in solving large problems. For this example, we consider a photolithography station that processes two different devices with four stepper machines. Each device consists of four layers. So we consider the lots that belong to the four layers of first device as first four products and the lots that belong to the four layers of second device as the next four products. So totally we have 8 products and 4 machines with 4 due dates. We consider each lot as a batch and the size of each batch depends on the product. General practice of wafer fab is to keep the lot size constant through out the fab and hence we assume that the batch size of each product is constant (25). The details of processing times, setup times and demands for each product are given in Tables 5.7 and 5.8 respectively.

Table 5.7: Processing times and setup times in Example 3

	Product							
Unit	I1	I2	I3	I4	I5	I6	I7	I8
J1	0.7/0.5				0.8/0.7*		0.7/0.6	
J2	0.85/0.5					0.95/0.6		1.0/0.65*
J3	0.85/0.55	0.9/0.6*	0.95/0.65				1.0/0.65	
J4			1.1/0.6	0.75/0.55		0.85/0.6*		

*unit is preoccupied with that particular product

We penalized the delay on each product in such a way that we get the finished lots as early as possible. The penalties on each product are also shown in Table 5.8. Instead of sequence-dependent setup times, we have unit-based setup times for this example. So we changed our model by modifying some constraints to remove the sequence-dependent setup times and to accommodate unit-based setup times. All units are available at the start of the time horizon and are set to process some products. We have taken K_1=3, K_2=6, K_3=9, K_4=12 to solve this problem.

Table 5.8: Demands and penalties at each due date for each product in Example 3

	Demands/penalties at each due date			
Product	D1	D2	D3	D4
I1	650/1	600/1	600/1	600/1
I2	600/1.25	600/1.25	600/1.25	600/1.25
I3	600/1.75	650/1.75	600/1.75	700/1.75
I4	600/2	600/2	600/2	700/2
I5	600/1	650/1	700/1	650/1
I6	600/1.25	650/1.25	600/1.25	700/1.25
I7	600/1.75	600/1.75	650/1.75	650/1.75
I8	700/2	600/2	650/2	700/2

Table 5.9: Model statistics for Example 3

Statistics	M1	M2
Constraints	1483	1219
Continuous	823	823
Binary	156	156
Integer	156	156
Nodes	6396	1017
Non-zeros	5734	5188
Iterations	129634	32823
CPU time (s)	39.5	12.3

Both our models able to solve this problem in short CPU time and the results are shown in Table 5.9. L&K model failed to solve this problem within reasonable amount of time (~10000 s) and hence we did not compare our work with L&K model in this example. RMIP and MILP values are zero in both models because we can meet all the demands with in the due dates. Surprisingly, M2 has performed better than M1 in this example and is evident from the results shown in Table 5.9. Because of the increase in the problem size, we think that the number of equations and non-zeros played an important role in model evaluation and gave M2 an edge over M1 in this example.

5.3.4 Example 4

Though Example 3 demonstrates that our formulation can solve large scale problems, it does not include sequence-based setup times. So, we illustrate the effectiveness of our work with this example that includes all the features of a general chemical batch plant.

Table 5.10: Batch size limits in Example 4

	Unit						
Product	1	2	3	4	5	6	7
1		100-200	100-200	100-200			
2		50-100					
3	100-200						
4			100-200				
5					100-200	100-200	
6	100-200						
7		100-200					
8							100-200
9							100-150
10				50-100			
11	50-100						
12					50-100	100-200	
13							100-200
14		100-200					

Table 5.11: Processing times of products on each unit in Example 4

Unit	Product													
	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14
J1			0.1			0.1					0.15			
J2	0.05	0.1					0.05							0.1
J3	0.1			0.1										
J4	0.1									0.05				
J5					0.1							0.05		
J6					0.1							0.1		
J7								0.15	0.1				0.1	

Table 5.12: Transition times between products in Example 4

Product	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1		1.00		0.50			0.50			0.50				0.50
2	0.50						0.40							0.30
3						0.20					0.15			
4	0.25													
5												0.30		
6			0.50								1.00			
7	0.60	1.00												0.50
8									0.50				1.00	
9								0.60					1.00	
10	1.00													
11			0.50			0.50								
12					1.00									
13								0.50	0.25					
14	1.00	.50					0.50							

A plant produces 14 products (I =14) with 7 units (J=7) in operating mode. As all the products cannot be processed on all the machines, the plant has problems in producing the products for customers within time. It is desired to schedule the products for the next 3 months. For the first four weeks, the schedule must be a detailed one. So the first four periods contain 7 days each and we fix 5th and 6th due dates as the end of 2nd and 3rd month respectively. The scheduling period comprises 6 due dates (d=6).

Table 5.13: Demands for each product at each due date in Example 4

Product	Due Date					
	D1	D2	D3	D4	D5	D6
I1	1000	1200	1200	1200	1000	800
I2	500	600	500	900	500	300
I3	500	500	900	500	400	800
I4	500	400	400	900	600	500
I5	800	700	800	900	1000	1000
I6	400	400	300	500	400	500
I7	500	500	600	400	500	400
I8	300	300	200	500	500	500
I9	200	400	300	400	300	500
I10	1000	800	800	800	1000	1000
I11	400	200	400	500	400	500
I12	600	600	500	800	500	500
I13	500	800	500	800	600	800
I14	500	500	500	800	600	500

There is no FPT_{ij} of the batches to be processed and hence the processing time of a batch consists of only VPT_{ij} that depends both on product and unit. The penalties on delay of each product are same and equal to one. All the units are available at the start of the horizon. The limits on batch size are shown in Table 5.10. The processing times of each product on each machine are listed in Table 5.11 and the transition times between different products are given in Table 5.12. Demands for each product at each due date are shown in Table 5.13. The minimum and maximum limits on number of batches on each unit are 1 and 100 respectively. We use K_1=3, K_2=6, K_3=9, K_4=12, K_5=15, K_6=18 to solve this problem.

The results for this example are shown in Table 5.14. Both our models needed less computational effort to solve this problem. As in Example 3, L&K model needs very large number of slots to solve the problem and hence could not solve the problem in reasonable amount of time (~10000 s). Hence, we do not compare our work with L&K in this example too. Demands are met within due dates in this example and hence both RMIP and MILP values are zero. Again both M1 and M2 competed well against each other even though model M2 slightly has an edge over model M1 in this example too. Again, this could be attributed to fewer equations and non-zeros in model M2.

Table 5.14: Model statistics for Example 4

Statistics	M1	M2
Constraints	4507	3859
Continuous	2820	2820
Binary	324	324
Integer	324	324
Nodes	1297	799
Iterations	25804	14044
Non-zeros	13426	16846
CPU time (s)	21.88	19.32

5.4 Remarks

(a) L&K used binary variables for transitions between batches that belong to different products. By doing this, binary variables in their model increase tremendously with the problem size and hence they could not solve bigger size problems. By using continuous variables for transitions between products on units in our model, we are able to reduce the number of binary variables.

(b) If we use a criterion to estimate the number of slots, we may overestimate the number of slots and there might be more number of binary variables. So in that sense it is better to use an iterative procedure to estimate the number of slots required. But then, we need to run the model many times to finalize the number of slots!

(c) To reduce the number of slots to schedule large number of batches, we introduced integer variables. By using integer variables, we not only reduced the binary variables but also could able to solve the scheduling problems with less computational effort. We also observed that the effect of integer variables on the performance of a model is relatively insignificant.

(d) Past scheduling work on such a process addressed the problem in terms of individual batches. The problem with such an approach is that it becomes impossible to schedule large number of batches waiting before the processing units. So in our work, by scheduling multiple batches in one slot, we require fewer slots and hence binary variables. Even though it introduces the integer variables, as shown in the results, integer variables have very less effect on CPU time.

(e) Both our models differ mainly in expressing the number of batches and hence amount produced in each slot. Model M2 tries to produce at least one batch in each useful slot where as model M1 takes whatever number of batches it requires. With this little change, it seems that there is not much difference between the performance of M1 and M2. But we cannot rule out the possibility of the dominance of one model against another as it is evident from Example 2 and Example 3. In Example 2, M1 performed better than M2 where as in Example 3 & Example 4, M2 showed better results than M1.

(f) Generally Big-*M* affects solution times very much irrespective of models. By introducing an objective free of big-*M*, we are able to compare our work with other models without big-*M* effect and hence could ensure a fair comparison. But even with tardiness as objective that has big-*M* constraint, our models have shown tremendous performance over other work and also they are more robust with respect to *M*-value.

5.5 Conclusions

We proposed two different MILP formulations for the short term scheduling of a multiproduct single stage batch plants with parallel units. The models can handle multiple orders per product and large number of batches without much computational effort. The concept of using more number of batches in a single slot is very novel and very effective. Two scheduling objectives were selected to compare our work with existing models. One objective is minimizing tardiness that involves big-*M* constraint and the other objective is minimizing delayed amount, which does not have any big-*M* constraint. With this, we could compare the models with and without the effect of big-*M*. In both cases, our models outperform the previous work with respect to the statistics (variables, constraints, non-zeros, nodes, iterations and CPU times). From the results, we conclude that our models are very robust with respect to big-*M* than the previous work.

Chapter 6

APPLICATION IN SEMICONDUCTOR MANUFACTURING

6.1 Scheduling in Semiconductor Manufacturing

Various processes involved in semiconductor manufacturing were discussed in detail in the previous chapter. Having conferred about the process, we now focus on the scheduling in semiconductor manufacturing. With considerable existing work in the modeling, simulation and optimization of actual physicochemical operations, it has been estimated that major improvements in the productivity of semiconductor industry lay in the higher-level systems engineering research involving the planning, scheduling, and optimization of semiconductor manufacturing. However, the semiconductor industry is quite different from the more traditional manufacturing operations such as assembly lines or job shops. Features such as re-entrancy, resource constraints and uncertainties make this industry a highly complex process system that is difficult to schedule. Modeling the entire suite of operations is still a daunted task, and we have been trying to focus on some specific, most critical, and bottleneck steps. One such step is the photolithography.

Photolithography is the most complex operation in wafer fabrication, and it requires the greatest precision. Photolithography is used to create multiple layers of circuit patterns on a chip. This area, where wafers are exposed using scanners or steppers, typically, comprises the bottleneck workstations. Also, the number of reticles (masks) available for a given layer of product type is limited. It is very important to develop schedules that ensure the maximum utilization of the equipment in this processing area. The main steps that a wafer has to undergo during a photolithography step are coating of the photo resist, exposure of the resist and the development of the resist. In the first step, the wafer is spun while the resist is deposited onto the wafer. The wafer is then baked to firm the photoresist and improve its adhesion to the wafer. The wafer is exposed to the UV light through the reticle that contains the pattern for a few chips.

This alignment and exposure step is repeated until the whole wafer surface is exposed. The wafer is then sent to the developing step, where the exposed photoresist is removed with a chemical solvent, and then it goes through final bake to ensure that the unexposed photo resist adheres to the wafer.

The tool sets used in photolithography are characterized by the function they perform and by their limits on critical dimensions. The tool sets used in the process are the scanners and steppers. Scanners are used for processing critical layers while steppers are used for processing non-critical layers. So all the units/machines are not identical in the photolithography station and we can process a particular layer only on a particular type of machine. This scheduling problem is similar to the one described in chapter 3 and there is some work in the literature on this type of problem.

But, in existing studies, the problem data are assumed to be deterministic. However, in real plants, parameters such as raw material availability, processing times, and market requirements vary with respect to time and are often subject to unexpected deviations. These uncertainties are common and can have undesirable short-term and long-term economic and feasibility implications. Therefore, the consideration of uncertainty in scheduling problems is of great importance in preserving plant feasibility and viability during operations. Although a large number of papers have addressed uncertainty in process design, much less attention has been devoted to the issue of uncertainty in process planning and scheduling, mainly because of the increased complexity of the deterministic problem.

Among the works on this subject appearing in the literature is that of Shah and Pantelides (1992) which addresses the problem of design of multipurpose batch plants. They considered different schedules for different sets of production requirements using a scenario-based approach and an approximate solution strategy. Pistikopoulos and Ierapetritou (1995) presented a two-stage stochastic programming formulation for the problem of batch plant design and operations under uncertainty. Straub and Grossmann (1993) proposed the idea of the stochastic flexibility index to evaluate the effect of uncertainty quantitatively. Bansal et al. (2002) proposed a parametric programming framework for the flexibility analysis design of linear systems, and in their subsequent work, they generalized and unified this approach to handle nonlinear systems.

The multiperiod planning and scheduling of multiproduct plants under demand uncertainty was addressed by Petkov and Maranas (1997). The stochastic elements in their proposed model are expressed with equivalent deterministic forms, resulting in a convex MINLP problem. Schmidt and Grossmann (1996) considered the optimal scheduling of new product testing tasks and reformulated the initial nonlinear, nonconvex disjunctive model as an MILP using different sets of simplifying assumptions that give rise to different models.

The uncertainties in planning and scheduling problems are generally described through probabilistic models. During the past decade, fuzzy set theory has been applied to scheduling optimization using heuristic search techniques. Balasubramanian and Grossmann (2002) developed MILP models for flowshop scheduling with uncertain processing times using discrete probability distribution. Later, the same authors (2003) proposed an alternative approach based on a fuzzy representation of uncertainty.

Vin and Ierapetritou (2001) proposed a multiperiod programming model to improve the schedule performance of batch plants under demand uncertainty. Recently, Jia and Ierapatritou (2004) proposed an integrated framework to address the issue of uncertainty in short-term scheduling. In their work, the idea of inference-based sensitivity analysis for MILP problems is utilized within a branch-and-bound solution framework to determine the importance of different parameters and constraints and to provide a set of alternative schedules for the range of uncertain parameters under consideration.

However, most of the existing works handle only a certain type of uncertain parameters. But in semiconductor manufacturing processes, uncertainty exists in processing times, demands and even in reentrant lots i.e. the number of lots that enter the stage for re-processing. So, we try to develop a MILP formulation that accounts for these uncertainties. In the next section, we describe the scheduling problem.

6.2 Problem Description

In a photolithography station, there are N lots to process on J machines. The problem that we address in this chapter can be defined as follows: Given a number of lots in a queue at a particular workstation and probabilistic scenarios, determine the expected total production of lots. The main features of the scheduling problem are:

a. There are different types of wafers in the system and each has to undergo many operations for different layers.

b. There are different types of machines available, which can be used for a particular operation. A machine may process one or more wafers. Let $\boldsymbol{J_i}$ denote the set of units that may process product *i*, and Let $\boldsymbol{I}_j$ denote the set of products that a unit *j* may process.

c. The type of flow is reentrant i.e. a lot can visit the photolithography station many times during its production.

d. The processing time of a lot on a machine is not constant and varies with respect to a given scenario. A scenario is a set of possible data of a process which may occur. Simply, it is a possible set of future events.

e. Transition from one device to another on any machine requires unit-based setup times.

f. Some units may not be available at the start of scheduling horizon, as they may need some time to process current batches. Similarly, it may not be possible to start processing some products at the start of horizon because of the lack of raw materials. To accommodate these situations, we use unit release times and job release times as the earliest times at which a machine or product may start processing.

Following assumptions are made while developing the MILP formulation:

1. We consider lots that belong to the same layer as batches belong to the same product. So each product represents one layer. If any lot that has processed for a layer returns to stepper machines then we consider it as a lot that belongs to another product. Implicitly we do not consider precedence relationships between adjacent layers.

2. We do not consider resource constraints. So we assume that masks are always available to units to process lots in sufficient quantity.

3. We consider all lots that belong to different layers of devices as the batches for different products. For example, two different devices that have four layers each are considered as eight different products.
4. The reentrant lots will join the system only at the end of a period and this is pre-determined.
5. The operation is non-preemptive and there are no resource constraints.

In this case, the lots at different stages of production are queued up in front of the stage. The issue then is which of these lots to process next. Another issue is the dynamic environment in which the lots enter and exit the system periodically.

Thus, the objectives are:

(i) determine how to allocate the scanners and steppers to different product types during their visit to photolithography station in a stochastic environment, and (ii) decide when to make changeovers for setting up the machines for different product types and different layers during their visits, in order to utilize the machines efficiently in the photolithography area as well as to increase the production of the facility. Thus, the above problem can be viewed as a sequencing problem with special characteristics and stochastic specifications.

6.3 Model Formulation

We develop the stochastic model using slot-based time representation. We use tri-index binary variables in this work.

$$Y_{ijk} = \begin{cases} 1 & \text{if machine } j \text{ processes product } i \text{ during slot } k \\ 0 & \text{otherwise} \end{cases}$$

We allow exactly one product in each slot on each machine. Therefore,

$$\sum_{i \in \boldsymbol{I}_j} Y_{ijk} = 1 \tag{6.1}$$

For modeling transitions (setups) between products in successive slots, we use a 0-1 continuous variable $X_{ii'jk}$ as follows.

$$X_{ii'jk} = \begin{cases} 1 & \text{if product } i' \text{ in slot } k \text{ and product } i \text{ in slot } (k-1) \text{ on unit } j \\ 0 & \text{otherwise} \end{cases}$$

And we linearize this exactly as,

$$Y_{ij(k-1)} = \sum_{i'} X_{ii'jk} \qquad k > 0 \tag{6.2}$$

$$Y_{ijk} = \sum_{i'} X_{i'ijk} \qquad k > 0 \tag{6.3}$$

The length of a slot consists of processing time of the lots that are assigned to the slot and unit-based setup time, if any. The length of each slot varies with each scenario. Let TE_{jkl} is the end time of the slot k on machine j in scenario l and it is defined as,

$$TE_{jkl} \geq TE_{j(k-1)l} + \sum_i (Y_{ijk} - X_{iijk}) ST_{ijl} + \sum_i N_{ijkl} PT_{ijl} \tag{6.4}$$

The limits on the number of lots that can be processed are given by,

$$N_{ijkl} \leq N_{ij}^U Y_{ijk} \tag{6.5}$$

$$N_{ijkl} \leq N_{ij}^U (Y_{ijk} - X_{iijs}) \tag{6.6}$$

Eq. 6.5 ensures that the number of lots of product i on machine j in slot k in scenario l will never be more than the limit on maximum number of lots, if the product i is assigned to machine j in slot k. Eq. 6.6 ensures that there is no production in null slots.

General practice of a fab is to keep the lot size constant. If B_{ijkl} is the number of wafers of product i processed on machine j in slot k in scenario l, then B_{ijkl} is defined as,

$$B_{ijkl} = BS_i N_{ijkl} \tag{6.7}$$

To push the slots to the end of the period, we use,

$$X_{iij(k+1)} \geq X_{iijk} \qquad K_{(d-1)} + 1 \leq k < K_d \tag{6.8}$$

There is storage for each product at the end of each period. There are wafers that join the system (*P(i)*) at the end of each period (supply) and also that got processed during the period (demand). So the number of lots that is available in a storage at the end of a period is given by,

$$SI_{idl} = SI_{i(d-1)l} + P_i - \sum_j \sum_{k=1}^{K_d} B_{ijkl} \tag{6.9}$$

The number of wafers that can be processed in the next period cannot be more than the available in the storage. Therefore,

$$SI_{i(d-1)l} \geq \sum_j \sum_{k>K_{d-1}}^{K_d} B_{ijkl} \tag{6.10}$$

Because of the storage policy at the end of each period for each product, slots that belong to a period must end on or before that due date. So we force the following constraint.

$$TE(j,k,l) \leq DD_d \qquad \forall k \leq K_d \tag{6.11}$$

The objective is to maximize the production of the facility, i.e. to maximize the number of lots or wafers from photolithography station.

$$PRO = \sum_i \sum_j \sum_k \sum_l B_{ijkl} w(l) wc(i) \tag{6.12}$$

w(l) and *wc(i)* are probability of each scenario *l* and weight coefficient of each product *i* respectively.

Note that we can obtain the deterministic model by removing the index *l* over all the parameters and variables. In the next section, we evaluate the model performance and the effect of uncertainty on the fab performance.

6.4 Model Evaluation

We consider a photolithography station that processes 2 devices. Each device consists of 4 layers and hence the total number of products is 8 (*I* = 8). So the lots waiting before the stepper machines in the photolithography station belong to one of the 8 products. There are 4 machines (*J* = 4) to process the lots in the station. There are product-machine combinations because all machines cannot process a given lot/product.

Table 6.1: Processing times and unit-based setup times of each product on each machine in each scenario

	Scenario 1				Scenario 2			
	Machines				Machines			
Product	J1	J2	J3	J4	J1	J2	J3	J4
I1	0.7/0.5	0.85/0.50	0.85/0.55		0.9/1.1	1.1/1.0	1.2/0.9	
I2			0.90/0.60*				1.0/1.0*	
I3			0.95/0.65	1.10/0.60			1.1/1.1	1.00/0.95
I4				0.75/0.65				0.90/1.00
I5	0.8/0.7*				1.2/1.0*			
I6		0.95/0.60		0.85/0.60*		1.0/1.0		1.10/1.00*
I7	0.7/0.6		1.00/0.65		1.0/0.9		0.8/1.0	
I8		1.00/0.65*				1.0/1.1*		

* Machine is setup initially for that particular product.

We consider two scenarios (*L* = 2) in which processing time and setup time varies with respect to scenario. In scenario 1, the processing times and setup times are at a lower limit where as in scenario 2, they are at a higher limit. The processing times and setup times are given in Table 6.1 for each scenario. There is equal probability for each scenario to occur (*W(l)* = 0.5). Initially there are 400 wafers of each product (*SI(I,D*0*)* = 400) and at the end of each period there are 300 wafers of each product (*P(i)* = 300) to join the system as reentrant material. There are four due dates (*D* = 4) and are given as *D*1 = 24 hr, *D*2 = 48 hr, *D*3 = 72 hr, *D*4 = 96 hr.

We have taken 12 slots (K = 12) to solve this problem and they are evenly distributed among the due dates as follows: K_1=3, K_2=6, K_3=9, K_4=12. We solved deterministic model for both the scenarios and the objective values are 15,600 and 14750 for scenario 1 and scenario 2 respectively. Then we solved the stochastic model that accounts for both the scenarios. As the processing times are different for each scenario, the schedules are different though the allotments of products to units are same for both the scenarios. The objective value (15079), as expected, is between the two values given by deterministic models.

As shown above, the objective value in stochastic model deviates considerably from deterministic values. Therefore, the deterministic model not only predicts the expected value of objective value wrongly but also can yield a sub-optimal solution in the face of uncertainty. The outcome depends on several factors such as the number of reentrant lots at the end of each period ($P(i)$), probability of each scenario ($w(l)$), importance of each product ($wc(i)$). One more interesting thing is the CPU time for the evaluation of the performance of all models. For the deterministic model with lower processing times (scenario 1), CPU time to solve the model is 1 s where as for scenario 2, it is about 1000 s. But for the stochastic model, CPU time required to solve the MILP model is about 650 s.

6.5 Conclusions

This work has addressed the problem of scheduling single stage batch plants with reentrant flows in stochastic environment. We developed one MILP formulation in which we implemented the uncertainties in the variables and parameters for photolithography station by modeling it as single stage batch plant. We solved the problem for two different scenarios in which the processing times and unit-based setup times vary with each scenario. Then we compared the objective value achieved by the stochastic model with the deterministic model. It clearly shows that the deterministic model predicts the expected value of objective wrongly in the face of uncertainty. But, further work is recommended to analyze the model with more scenarios and with more variables. We expect that the model size increases rapidly with the number of scenarios and hence a powerful algorithm is needed to aid the model in analyzing the system with uncertainties.

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

Scheduling of single-stage noncontinuous multiproduct chemical processes with non-identical units/lines was addressed in this work. Four different slot-based Mixed Integer Linear Programming (MILP) formulations (semicontinuous units-no due date, multiple due dates; batch units-no due date, multiple due dates) were presented. We were able to develop the models without idle/null product and hence could reduce the number of binary variables involved. We also developed corresponding event-based formulations for all the cases. Later, slot-based models were compared with event-based models and the results put slot-based models ahead of event-based models. Especially RMIP values were in favor of slot-based models. Furthermore, event-based models could not solve some scheduling problems where as slot-based models solved those problems without taking much computational effort. Also, we observed that decoupling of tasks from units in a mathematical formulation cannot reduce the number of binary assignment variables.

Next, a novel way of scheduling multiproduct single stage batch plants with non-identical units was addressed. In this work, we developed a mathematical model to schedule multiple batches using slot-based continuous-time representation. Our model allows the scheduling of multiple batches of the same type of product in a single slot and hence is able to handle larger size problems. Although our model introduced integer variables, it reduced the binary variables considerably, and hence solved the scheduling problems faster. Numerical tests have shown that the integer variables as opposed to binary variables have much less effect on the model performance. Also, we showed that our models are very robust with respect to Big-M value.

Finally, we addressed the application of above work in the scheduling of photolithography station in semiconductor industry with stochastic data. One MILP formulation was developed in which we implemented the uncertainties in the variables and parameters for photolithography station by modeling it as a single stage batch plant. We solved the problem for two different scenarios in which the processing times and unit-based setup times vary with each scenario. Then we compared the objective value achieved by the stochastic model with the objective value obtained with the deterministic model. We showed that the deterministic model predicts the expected value of objective wrongly in the face of uncertainty.

7.1 Recommendations

Even though the comparison between slot-based and event-based models has considered reasonable size examples, bigger size examples are recommended to draw comparison. Also, in the present comparison, to keep the comparison simple we did not consider the transition times. Comparison may be done after incorporating transition times in the event-based. And, a comparison is recommended between the slot-based and the predecessor-successor based models for semicontinuous processes.

In chapter 5, our methodology was able to solve larger size problems where one can satisfy the demands on or before the due dates. Though our models can solve smaller size problems where one cannot satisfy the demands within the due dates, it failed to solve larger size problems for such scenarios. Further work is needed to solve the larger size, industrial scale problems in which we cannot meet the demands on or before the due dates. Some of the constraints may be rewritten in a slightly different manner or new constraints may be added, which may actually improve the effectiveness of these models.

Furthermore, we addressed only single-stage processes in this work. Ideas of this work can be applied to multi stage processes.

As mentioned in chapter 6, deterministic models are not useful when some uncertainty exists in model variables and parameters. Though we have studied this issue, further work is recommended to analyze the model with more scenarios and more variable parameters. We expect that the model size increases rapidly with the number of scenarios and hence a powerful algorithm is needed to aid the model in analyzing the system with uncertainties.

REFERENCES

1. Balasubramanian, J. and I. E. Grossmann. A Novel Branch and bound algorithm for scheduling flowshop plants with uncertain processing times, Comput. Chem. Eng., *26*, pp.41-57. 2002.

2. Balasubramanian, J. and I. E. Grossmann. Scheduling Optimization under Uncertainty- an Alternate Approach, Comput. Chem. Eng., *27*, pp.469-470. 2003.

3. Bansal, V., J. D. Perkins and E. N. Pistikopoulos. Flexibility Analysis and Design using a Parametric Programming Framework, AIChE Journal, *48*, pp.2851-2868. 2002.

4. Cerda, J., G. P. Henning and I. E. Grossmann. A Mixed-Integer Linear Programming Model for Short-Term Scheduling of Single-Stage Multi-product Batch Plants with Parallel Lines, Ind. Eng. Chem. Res., *36(5)*, pp.1695-1707. 1997.

5. Chen, C., C. Liu, X. Feng and H. Shao. Optimal Short-Term Scheduling of Multiproduct Single-Stage Batch Plants with Parallel Lines, Ind. Eng. Chem. Res., *41*, pp. 1249-1260. 2002.

6. Gupta, S. and I. A. Karimi. An improved MILP formulation for scheduling multiproduct, multistage batch plants, Ind. Eng. Chem. Res., *42(11)*, pp.2365-2380. 2003.

7. Hui, C. and A. Gupta. A Bi-Index Continuous-Time Mixed-Integer Linear Programming Model for Single-Stage Batch Scheduling with Parallel Units, Ind. Eng. Chem. Res., *40*, pp. 5960-5967. 2001.

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

9. Ierapetritou, M. G., T. S. Hene and C. A. Floudas. Effective Continuous-Time Formulation for Short-Term Scheduling. 3. Multiple Intermediate Due Dates, Ind. Eng. Chem. Res., *38*, pp.3446-3461. 1999.

10. Jia, Z. and M. G. Ierapetritou. Short-term Scheduling under Uncertainty using MILP Sensitivity Analysis, Ind. Eng. Chem. Res., *43*, pp.3782-3791. 2004.

11. Karimi, I. A. and C. M. McDonald. Planning and scheduling of parallel semicontinuous processes. 2. Short-term scheduling, Ind. Eng. Chem. Res., *36*, pp.2701-2714. 1997.

12. Kim, S., S. Yea and B. Kim. Shift Scheduling for Steppers in the Semiconductor Wafer Fabrication Process, IIE Transactions, *34(2)*, pp.167-177. 2002.

13. Kondili, E., C. C. Pantelides and R. W. H. Sargent. A General Algorithm for Short-term Scheduling of Batch Plants-I.MILP Formulation. Comput. Chem. Eng., *17*, pp.211-227. 1993.

14. Lamba, N., I. A. Karimi and A. Bhalla. Scheduling a single-product reentrant process with uniform processing times, Ind. Eng. Chem. Res., *39(11)*, pp.4203-4214. 2000.

15. Lamba, N. and I. A. Karimi. Scheduling Parallel Production Lines with Resource Constraints. 1. Model Formulation, Ind. Eng. Chem. Res., *41(4)*, pp.779-789. 2002a.

16. Lamba, N. and I. A. Karimi. Scheduling Parallel Production Lines with Resource Constraints. 2. Decomposition Algorithm, Ind. Eng. Chem. Res., *41(4)*, pp.790-800. 2002b.

17. Lim, M. F. and I. A. Karimi. A Slot-based Formulation for Single-Stage Multi-Product Batch Plants with Multiple Orders per Product, Ind. Eng. Chem. Res., *42(9)*, pp.1914-1924. 2003a.

18. Lim, M. F. and I. A. Karimi. Resource-constrained Scheduling of Parallel Production Lines Using Asynchronous slots, Ind. Eng. Chem. Res., *42(26)*, pp.6832-6842. 2003b.

19. Maravelias, C. T. and I. E. Grossmann. New General Continuous-Time State-Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants, Ind. Eng. Chem. Res., *42*, pp.3056-3074. 2003.

20. Mendez, C.A., G. P. Henning and J. Cerda. Optimal Scheduling of Batch Plants Satisfying Multiple Product Orders with Different Due Dates, Comput. Chem. Eng., *24*, pp.2223-2245. 2000.

21. Petkov, S. B. and C. D. Maranas. Multiperiod planning and scheduling of multiproduct batch plants under demand uncertainty, Ind. Eng. Chem. Res., *36*, pp.4864-4881. 1997.

22. Pinto, J. M. and I. E. Grossmann. Optimal Cyclic Scheduling of Multistage Continuous Multiproduct Plants, Comput. Chem. Eng., *18*, pp. 797-816. 1994.

23. Pinto, J. M. and I. E. Grossmann. Assignment and Sequencing Models for the Scheduling of Process Systems, Ann. Oper. Res., *81*, pp.433-466. 1998.

24. Pistikopoulos, E. N. and M. G. Ierapetritou. Novel Approach for Optimal Process Design under Uncertainty, Comput. Chem. Eng., *19*, pp.1089-1110. 1995.

25. Reklaitis, G. V. Perspectives on Scheduling and Planning of Process Operations, Proceedings of the 4th International Symposium on Process Systems Engineering, Montebello, Canada, 1991.

26. Sahinidis, N. V. and I. E. Grossmann. MINLP Model for Cyclic Multiproduct Scheduling on Continuous Parallel Lines, Comput. Chem. Eng., *15(2)*, pp.85-103. 1991.

27. Schmidt, C. and I. E. Grossmann. Optimization models for the scheduling of testing tasks in new product development. Ind. Eng. Chem. Res., *35*, pp.3498-3510. 1996.

28. Shah, N. and C. C. Pantelides. Design of Multipurpose Batch Plants with Uncertain Production Requirements, Ind. Eng. Chem. Res., *31*, pp.1325-1337. 1992.

29. Shah, N., C. C. Pantelides and R. W. H. Sargent. A General Algorithm for Short-Term Scheduling of Batch Operations – 2. Computational Issues, Comput. Chem. Eng., *17*, pp. 229-244. 1993.

30. Straub, D. A. and I. E. Grossmann. Design Optimization of stochastic flexibility, Comput. Chem. Eng., *17*, pp.339-354. 1993.

31. Sundaramoorthy, A. and I. A. Karimi. A Simpler Better Slot-based Continuous-time Formulation for Short-term Scheduling in Multipurpose Batch Plants, submitted to Ind. Eng. Chem. Res. 2004.

32. Vin, J. P. and M. G. Ierapetritou. Robust short-term scheduling of multiproduct batch plants under demand uncertainty, Ind. Eng. Chem. Res., *40*, pp.4543-4554. 2001.

APPENDIX A: Chapter 4

A.1: GAMS file for Example 1

```
SETS
I        products /I1*I8/
I1(I)    products /I1*I8/
I2(I1)   products /I1*I8/
I3(I1)   products /I1*I8/
I4(I1)   products /I1*I8/
J        units /J1*J3/
J1(J)    units /J1*J3/
D        duedates/D1*D4/
D1       duedates/D1*D4/
K        slots /K1*K3/
KA(K) slots /K1*K3/

IJ(I,J)          products that can be produced on line J
/(I1*I3,I8).J1,(I2*I4,I7).J2,(I4*I7).J3/
I1J(I1,J)        products that can be produced on line J
/(I1*I3,I8).J1,(I2*I4,I7).J2,(I4*I7).J3/
JI(J,I)          lines that can produce product I
/(J1).I1,(J1*J2).I2,(J1*J2).I3,(J2*J3).I4,J3.I5,J3.I6,(J2*J3).I7,J1.I8/
JI1(J,I1)        lines that can produce product I
/(J1).I1,(J1*J2).I2,(J1*J2).I3,(J2*J3).I4,J3.I5,J3.I6,(J2*J3).I7,J1.I8/

TABLE MPL(I,J) MIN PROD LENGTH OF I ON J
     J1   J2   J3
I1   10
I2   12   15
I3   12   10
I4        15   10
I5             12
I6             15
I7        10   10
I8   10

TABLE RU(I,J) MAX RATE OF PRODUCTION OF I ON J
     J1   J2   J3
I1   20
I2   20   25
I3   25   20
I4        20   20
I5             25
I6             20
I7        15   15
```

```
I8    15

SCALAR
H /50/;

PARAMETERS
DE(I) DEMAND/I1*I8 400/;

VARIABLES
MS;
BINARY VARIABLES Y;
POSITIVE VARIABLES RL,XX;

EQUATIONS
ASSIGN1(J,K)        Assignment
EXX1(I,J,K)         Transition1
EXX2(I,J,K)         Transition2
MAKESPAN(J)         Makespan
LIMRL1(I,J,K)       Limit on run length1
LIMRL2(I,J,K)       Limit on run length2
EX4(I,J,K)          Extra slots
DEMAND(I)           Demand;

ASSIGN1(J,K).. SUM(I$IJ(I,J),Y(I,J,K))=E=1;

EXX1(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K-1)=E=sum(I1$I1J(I1,J),XX(I,I1,J,K));

EXX2(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K)=E=sum(I1$I1J(I1,J),XX(I1,I,J,K));

EX4(I,J,K)$(ord(k) gt 2).. XX(I,I,J,K+1)=G=XX(I,I,J,K);

MAKESPAN(J).. SUM(K,SUM(I$IJ(I,J),RL(I,J,K)))=L=MS;

LIMRL1(I,J,K)$IJ(I,J)..    RL(I,J,K)=G=MPL(I,J)*(Y(I,J,K)-XX(I,I,J,K)$(ORD(K)    GT
2));

LIMRL2(I,J,K)$IJ(I,J).. RL(I,J,K)=L=H*(Y(I,J,K)-XX(I,I,J,K)$(ORD(K) GT 2));

DEMAND(I).. SUM(J$JI(J,I),SUM(K,RU(I,J)*RL(I,J,K)))=G=DE(I);

MODEL SC/ALL/;
Y.UP(I,J,K)=1;
XX.UP(I,I1,J,K)=1;

OPTION SOLPRINT=OFF;
OPTION OPTCR=0.0001;
```

```
SOLVE SC USING MIP MINIMIZING MS;

DISPLAY Y.L,RL.L,MS.L;
```

A.2: GAMS file for Example 2

```
SETS
I         products /I1*I8/
I1(I)     products /I1*I8/
I2(I1)    products /I1*I8/
I3(I1)    products /I1*I8/
I4(I1)    products/I1*I8/
J         units /J1*J3/
J1(J)     units /J1*J3/
D         duedates/D1*D4/
D1        duedates/D1*D4/
K         slots /K0*K8/
KA(K) slots /K0*K8/

IJ(I,J)            products that can be produced on line J
/(I1*I3,I8).J1,(I2*I4,I7).J2,(I4*I7).J3/
I1J(I1,J)          products that can be produced on line J
/(I1*I3,I8).J1,(I2*I4,I7).J2,(I4*I7).J3/
JI(J,I)            lines that can produce product I
/(J1).I1,(J1*J2).I2,(J1*J2).I3,(J2*J3).I4,J3.I5,J3.I6,(J2*J3).I7,J1.I8/
JI1(J,I1)          lines that can produce product I
/(J1).I1,(J1*J2).I2,(J1*J2).I3,(J2*J3).I4,J3.I5,J3.I6,(J2*J3).I7,J1.I8/


TABLE BSMIN(J,I) Batch size of I on J
    I1  I2    I3    I4   I5   I6   I7   I8
J1  50 100 100                          100
J2      100 100  80              100
J3              100  50  100 50

TABLE BSMAX(J,I) Batch size of I on J
     I1   I2    I3   I4    I5    I6    I7   I8
J1  100 140 150                             200
J2        150 150 120              150
J3                     150  100 200 100

TABLE FPT(J,I) PROCESSING TIME OF I ON J
    I1  I2  I3   I4   I5   I6   I7   I8
J1  3   2   3                        2
J2      2   3    2              1
```

```
J3              3    3   2   2

TABLE PT(J,I) PROCESSING TIME OF I ON J
     I1   I2   I3   I4   I5   I6   I7   I8
J1  .15  .16  .16                       .15
J2       .17  .16  .15            .17
J3                 .14  .16  .17  .18

PARAMETER
RTU(J) RELEASE TIME OF UNIT J
/J1 0,J2 0,J3 0/
DE(I)  demand
/I1 400,I2 250,I3 400,I4 500,I5 300,I6 250,I7 200,I8 300/

SCALAR
M large positive number/1000/
H HORIZON /96/

VARIABLES
Y(I,J,K)         assignment of order I to slot K
MS               makespan
OBJ              objective;
POSITIVE VARIABLES XX,B;
BINARY VARIABLE Y;
EQUATIONS
EX1(J,K)              Assignments
EXX1(I,J,K)           Transition1
EXX2(I,J,K)           Transition2
DEMAND(I)             Demand
EX4(I,J,K)            Extra slots
BATCH1(I,J,K)         Batch limit 1
BATCH2(I,J,K)         Batch limit 2
MAKESPAN(J)           Objective;


EX1(J,K).. SUM(I$IJ(I,J),Y(I,J,K))=E=1;

EXX1(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K-1)=E=sum(I1$I1J(I1,J),XX(I,I1,J,K));

EXX2(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K)=E=sum(I1$I1J(I1,J),XX(I1,I,J,K));

EX4(I,J,K)$(ord(k) gt 2).. XX(I,I,J,K+1)=G=XX(I,I,J,K);

BATCH1(I,J,K)$(IJ(I,J)   AND   (ORD(K)   GT   1))..   BSMIN(J,I)*(Y(I,J,K)-
XX(I,I,J,K)$(ORD(K) GT 2))=L=B(I,J,K);
```

```
BATCH2(I,J,K)$(IJ(I,J) AND (ORD(K) GT 1)).. BSMAX(J,I)*(Y(I,J,K)-
XX(I,I,J,K)$(ORD(K) GT 2))=G=B(I,J,K);

DEMAND(I).. DE(I)=L=SUM(J$JI(J,I),SUM(K$(ORD(K) GT 1),B(I,J,K)));

MAKESPAN(J)..MS=G=SUM(K$(ORD(K)GT
1),SUM(I$IJ(I,J),(FPT(J,I)*Y(I,J,K)+PT(J,I)*B(I,J,K))));


MODEL BATCH /all/;
BATCH.iterlim=100000000;
BATCH.reslim=100000000;
Y.UP(I,J,K)=1;
XX.UP(I,I1,J,K)=1;
Y.FX("I3","J2","K0")=1;
Y.FX("I1","J1","K0")=1;
Y.FX("I7","J3","K0")=1;
Y.FX("I1","J2","K0")=0;
Y.FX("I2","J1","K0")=0;
Y.FX("I2","J2","K0")=0;
Y.FX("I3","J1","K0")=0;
Y.FX("I4","J2","K0")=0;
Y.FX("I4","J3","K0")=0;
Y.FX("I5","J3","K0")=0;
Y.FX("I6","J3","K0")=0;
Y.FX("I7","J2","K0")=0;
Y.FX("I8","J1","K0")=0;

*option optca = .01;
option optcr = 0.0001;
OPTION SOLPRINT=OFF;
SOLVE BATCH USING MIP MINIMIZING MS;

DISPLAY Y.L,XX.L,MS.L,B.L;
```

A.3: GAMS file for Example 3

```
SETS
I       products /I1*I8/
I1(I)   products /I1*I8/
I2(I1)  products /I1*I8/
I3(I1)  products /I1*I8/
I4(I1)  products /I1*I8/
J       units /J1*J3/
J1(J)   units /J1*J3/
D       duedates/D1*D3/
```

```
D1      duedates/D1*D3/
K       slots /K0*K9/
KA(K) slots /K0*K9/

IJ(I,J)          products that can be produced on line J
/(I1*I3,I8).J1,(I2*I4,I7).J2,(I4*I7).J3/
I1J(I1,J)        products that can be produced on line J
/(I1*I3,I8).J1,(I2*I4,I7).J2,(I4*I7).J3/
JI(J,I)          lines that can produce product I
/(J1).I1,(J1*J2).I2,(J1*J2).I3,(J2*J3).I4,J3.I5,J3.I6,(J2*J3).I7,J1.I8/
JI1(J,I1)        lines that can produce product I
/(J1).I1,(J1*J2).I2,(J1*J2).I3,(J2*J3).I4,J3.I5,J3.I6,(J2*J3).I7,J1.I8/
KD(K,D)
/(K1*K3).D1,(K1*K6).D2,(K1*K9).D3/
FIRSTK(K)
/K1,K4,K7/
LKD(K,D)
/(K1*K2).D1,(K4*K5).D2,(K7*K8).D3/;

TABLE MPL(I,J) MIN PROD LENGTH OF I ON J
    J1   J2   J3
I1  10
I2  12   15
I3  12   10
I4       15   10
I5            12
I6            15
I7       10   10
I8  10

TABLE RU(I,J) MAX RATE OF PRODUCTION OF I ON J
    J1   J2   J3
I1  20
I2  20   25
I3  25   20
I4       20   20
I5            25
I6            20
I7       15   15
I8  15

TABLE Q(I,D1) Amount of product I required at duedate D
   D1   D2   D3
I1 100  100  100
I2 100  100  100
I3 100  200  100
```

```
I4  200  100  200
I5  200  300  100
I6  150  200  200
I7  100  100  100
I8  200  200  100

TABLE CT(I1,I) Transition time bet I1&I on J
   I1   I2   I3
I1  0.0  1.5  1.6
I2  5.1  0.0  1.3
I3  1.6  2.3  0.0


TABLE ALPHA(I,D) Weight coefficient for delay of product I at due date D
   D1   D2   D3
I1  1    1    1
I2  1    1    1
I3  1    1    1
I4  1    1    1
I5  1    1    1
I6  1    1    1
I7  1    1    1
I8  1    1    1


PARAMETER
RTU(J) RELEASE TIME OF UNIT J
/J1 0,J2 0,J3 0/
DD(D) due date D(hr)
/D1 24,D2 48,D3 72/

SCALAR
M /100/;

VARIABLES
OBJ;
BINARY VARIABLES Y;
POSITIVE VARIABLES RL,XX,DE,TE;

EQUATIONS
ASSIGN1(J,K)
EXX1(I,J,K)
EXX2(I,J,K)
LIMRL1(I,J,K)
LIMRL2(I,J,K)
LIMRL3(I,J,K)
```

```
EX4(I,J,K,D)
RT(J)
PROD4b(J,K)
DELAY(I,J,K,D)
DEMAND(I,D)
OBJECTIVE;

ASSIGN1(J,K).. SUM(I$IJ(I,J),Y(I,J,K))=E=1;

EXX1(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K-1)=E=sum(I1$I1J(I1,J),XX(I,I1,J,K));

EXX2(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K)=E=sum(I1$I1J(I1,J),XX(I1,I,J,K));

LIMRL1(I,J,K)$(IJ(I,J) AND (ORD(K) GT 1)).. RL(I,J,K)=G=MPL(I,J)*(Y(I,J,K)-
XX(I,I,J,K)$(NOT FIRSTK(K)));

LIMRL2(I,J,K)$(IJ(I,J) AND (ORD(K) GT 1)).. RL(I,J,K)=L=M*Y(I,J,K);

LIMRL3(I,J,K)$(IJ(I,J) AND (ORD(K) GT 1)).. RL(I,J,K)=L=M*(Y(I,J,K)-
XX(I,I,J,K)$(NOT FIRSTK(K)));

EX4(I,J,K,D)$(LKD(K,D) and ord(k) gt 2).. XX(I,I,J,K+1)=G=XX(I,I,J,K);

RT(J).. TE(J,"K0")=E=RTU(J);

PROD4b(J,K)$(ord(K) GT 1).. TE(J,K)=G=TE(J,K-1)+SUM(I$IJ(I,J),RL(I,J,K));

DEMAND(I,D)..
SUM(J$JI(J,I),SUM(K$KD(K,D),RU(I,J)*RL(I,J,K)))=G=SUM(D1$(ORD(D1) LE
ORD(D)),Q(I,D1));

DELAY(I,J,K,D)$(JI(J,I) AND KD(K,D)).. DE(I,D)=G=TE(J,K)-DD(D)-M*(1-
Y(I,J,K));

OBJECTIVE.. OBJ=E=SUM(D,SUM(I,ALPHA(I,D)*DE(I,D)));

MODEL SC/ALL/;
Y.UP(I,J,K)=1;
*X.UP(I,I1,J,K)=1;
Y.FX("I3","J2","K0")=1;
Y.FX("I1","J1","K0")=1;
Y.FX("I7","J3","K0")=1;
Y.FX("I1","J2","K0")=0;
Y.FX("I2","J1","K0")=0;
Y.FX("I2","J2","K0")=0;
Y.FX("I3","J1","K0")=0;
```

```
Y.FX("I4","J2","K0")=0;
Y.FX("I4","J3","K0")=0;
Y.FX("I5","J3","K0")=0;
Y.FX("I6","J3","K0")=0;
Y.FX("I7","J2","K0")=0;
Y.FX("I8","J1","K0")=0;
OPTION SOLPRINT=OFF;
OPTION OPTCR=0.0001;

SOLVE SC USING MIP MINIMIZING OBJ;

DISPLAY TE.L,Y.L,RL.L,DE.L,OBJ.L;
```

A.4: GAMS file for Example 4

```
SETS
I        products /I1*I8/
I1(I)    products /I1*I8/
I2(I1)   products /I1*I8/
I3(I1)   products /I1*I8/
I4(I1)   products /I1*I8/
J        units /J1*J3/
J1(J)    units /J1*J3/
D        duedates/D1*D3/
D1       duedates/D1*D3/
K        slots /K0*K9/
KA(K) slots /K0*K9/


IJ(I,J)          products that can be produced on line J
/(I1*I3,I8).J1,(I2*I4,I7).J2,(I4*I7).J3/
I1J(I1,J)        products that can be produced on line J
/(I1*I3,I8).J1,(I2*I4,I7).J2,(I4*I7).J3/
JI(J,I)          lines that can produce product I
/(J1).I1,(J1*J2).I2,(J1*J2).I3,(J2*J3).I4,J3.I5,J3.I6,(J2*J3).I7,J1.I8/
JI1(J,I1)        lines that can produce product I
/(J1).I1,(J1*J2).I2,(J1*J2).I3,(J2*J3).I4,J3.I5,J3.I6,(J2*J3).I7,J1.I8/
KD(K,D)
/(K1*K3).D1,(K1*K6).D2,(K1*K9).D3/
LKD(K,D)
/(K1*K2).D1,(K4*K5).D2,(K7*K8).D3/
FIRSTK(K)
/K1,K4,K7/
```

TABLE BSMIN(J,I) Batch size of I on J

	I1	I2	I3	I4	I5	I6	I7	I8
J1	50	100	100					100
J2		100	100	80			100	
J3				100	50	100	50	

TABLE BSMAX(J,I) batch size of I on J

	I1	I2	I3	I4	I5	I6	I7	I8
J1	100	140	150					200
J2		150	150	120			150	
J3				150	100	200	100	

TABLE FPT(J,I) PROCESSING TIME OF I ON J

	I1	I2	I3	I4	I5	I6	I7	I8
J1	3	2	3					2
J2		2	3	2			1	
J3				3	3	2	2	

TABLE PT(J,I) PROCESSING TIME OF I ON J

	I1	I2	I3	I4	I5	I6	I7	I8
J1	.15	.16	.16					.15
J2		.17	.16	.15			.17	
J3				.14	.16	.17	.18	

TABLE Q(I,D1) Amount of product I required at duedate D

	D1	D2	D3
I1	100	100	100
I2	100	100	100
I3	100	200	100
I4	50	100	100
I5	100	100	100
I6	100	200	100
I7	100	100	100
I8	50	100	100

TABLE ALPHA(I,D) Weight coefficient for delay of product I at due date D

	D1	D2	D3
I1	1	1	1
I2	1	1	1
I3	1	1	1
I4	1	1	1
I5	1	1	1
I6	1	1	1
I7	1	1	1
I8	1	1	1

```
PARAMETER
RTU(J) RELEASE TIME OF UNIT J
/J1 0,J2 0,J3 0/
DD(D)
/D1 50,D2 100,D3 200/

SCALAR
M large positive number/500/


VARIABLES
Y(I,J,K)  assignment of order I to slot K
MS        makespan
OBJ       objective;
POSITIVE VARIABLES XX,B,DE,TE;
BINARY VARIABLE Y;
EQUATIONS
PROD4b(J,K)         Timing constraint
DELAY(I,J,K,D)      Delay order constraint
EX1(J,K)            Assignments
EXX1(I,J,K)         Transition1
EXX2(I,J,K)         Transition2
EX4(I,J,K,D)        Extra slots
RT(J)               Release times
BATCH1(I,J,K)       Batch limit 1
BATCH2(I,J,K)       Batch limit 2
PROD2(I,D)          DEMAND
OBJECTIVE           Minimize tardiness or makespan or tardy orders;



DELAY(I,J,K,D)$(JI(J,I) and KD(K,D)).. DE(I,D)=G=TE(J,K)-DD(D)-M*(1-Y(I,J,K));

RT(J).. TE(J,"K0")=E=RTU(J);

EX1(J,K).. SUM(I$IJ(I,J),Y(I,J,K))=E=1;

EXX1(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K-1)=E=sum(I1$I1J(I1,J),XX(I,I1,J,K));

EXX2(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K)=E=sum(I1$I1J(I1,J),XX(I1,I,J,K));

EX4(I,J,K,D)$(LKD(K,D) and ord(k) gt 2).. XX(I,I,J,K+1)=G=XX(I,I,J,K);
```

```
PROD4b(J,K)$((ord(K) GT 1)).. TE(J,K)=G=TE(J,K-1)+SUM(I$IJ(I,J),(FPT(J,I)*Y(I,J,K)+PT(J,I)*B(I,J,K)));

PROD2(I,D)..
SUM(J$JI(J,I),SUM(K$(KD(K,D) AND ORD(K) GT 1),B(I,J,K)))
=G=SUM(D1$(ord(D1) LE ord(D)),Q(I,D1));

BATCH1(I,J,K)$(IJ(I,J) AND (ORD(K) GT 1)).. BSMIN(J,I)*(Y(I,J,K)-XX(I,I,J,K)$(NOT FIRSTK(K)))=L=B(I,J,K);

BATCH2(I,J,K)$(IJ(I,J) AND (ORD(K) GT 1)).. BSMAX(J,I)*(Y(I,J,K)-XX(I,I,J,K)$(NOT FIRSTK(K)))=G=B(I,J,K);

OBJECTIVE.. OBJ=E=SUM(D,SUM(I,ALPHA(I,D)*DE(I,D)));



MODEL BATCH /all/;
BATCH.iterlim=100000000;
BATCH.reslim=100000000;
Y.UP(I,J,K)=1;
XX.UP(I,I1,J,K)=1;
Y.FX("I3","J2","K0")=1;
Y.FX("I1","J1","K0")=1;
Y.FX("I7","J3","K0")=1;
Y.FX("I1","J2","K0")=0;
Y.FX("I2","J1","K0")=0;
Y.FX("I2","J2","K0")=0;
Y.FX("I3","J1","K0")=0;
Y.FX("I4","J2","K0")=0;
Y.FX("I4","J3","K0")=0;
Y.FX("I5","J3","K0")=0;
Y.FX("I6","J3","K0")=0;
Y.FX("I7","J2","K0")=0;
Y.FX("I8","J1","K0")=0;

*option optca = .01;
option optcr = 0.0001;
OPTION SOLPRINT=OFF;
SOLVE BATCH USING MIP MINIMIZING OBJ;

DISPLAY Y.L,XX.L,OBJ.L,TE.L,DE.L,B.L;
```

APPENDIX B: Chapter 5

B.1: GAMS file for Example 2 for Model M1

```
SETS
I        products /I1*I4/
I1(I)    products /I1*I4/
I2(I1)   products /I1*I4/
I3(I1)   products /I1*I4/
D        duedates /D1*D4/
D1(D)    duedates /D1*D4/
J        units /J1*J3/
J1(J)    units /J1*J3/
K        slots /K0*K5/
KA(K) slots /K0*K5/

IJ(I,J)          products that can be produced on line J
/(I2*I4).J1,(I1,I3*I4).J2,(I1*I2).J3/
I1J(I1,J)        products that can be produced on line J
/(I2*I4).J1,(I1,I3*I4).J2,(I1*I2).J3/
I2J(I2,J)        products that can be produced on line J
/(I2*I4).J1,(I1,I3*I4).J2,(I1*I2).J3/
ID(I,D)          products due on D
/(i1*i3).D1,(I1*I4).D2,(I1*I4).D3,(I1*I2,I4).D4/
ID1(I,D1)        products due on D
/(i1*i3).D1,(I1*I4).D2,(I1*I4).D3,(I1*I2,I4).D4/
JI(J,I)          lines that can produce product I
/(J2*J3).I1,(J1,J3).I2,(J1*J2).I3,(J1*J2).I4/
JI1(J,I1)        lines that can produce product I
/(J2*J3).I1,(J1,J3).I2,(J1*J2).I3,(J1*J2).I4/
KD(K,D)          slots to take place before duedate D
/(K1).D1,(K1*K2).D2,(K1*K3).D3,(K1*K5).D4/
NKD(K,D)         slots to take place except first slot before duedate D
/(K5).D4/
LKD(K,D)         slots to take place except last slot before duedate D
/(K1).D1,(K2).D2,(K3).D3,(K4).D4/
FIRSTK(K)
/K1*K4/

TABLE BSMIN(J,I) Min. Batch size of I on J
     I1   I2   I3   I4
J1        100  100 150
J2  100        100 100
J3  140 80
```

```
TABLE BSMAX(J,I) Max. Batch size of I on J
     I1    I2    I3    I4
J1         140  150  200
J2  120         120  150
J3  160 120

TABLE FPT(J,I) Fixed Processing Time of I on J
   I1  I2  I3  I4
J1      3   2   2.5
J2  5       2   3
J3  5   4

TABLE PT(J,I) Variables Processing Time of I on J
     I1   I2     I3    I4
J1         0.15 .17   0.15
J2  .15         .17 0.155
J3  .145 .155

TABLE CT(I1,I) Transition time bet I1&I on J
     I1   I2   I3   I4
I1  0.0  1.5  1.6  2.7
I2  5.1  0.0  1.3  4.8
I3  1.6  2.3  0.0  1.4
I4  1.0  2.5  2.1  0.0

TABLE Q(I,D) Amount of product I required at duedate D
    D1   D2   D3   D4
I1  50   100   100  100
I2  100  100  100  200
I3  50   100   100
I4       200   100  100

TABLE ALPHA(I,D) Weight coefficient for delay of product I at due date D
   D1   D2   D3   D4
I1  1    1     1     1
I2  1    1     1     1
I3  1    1     1     1
I4  1    1     1     1

PARAMETER
DD(D) due date D(hr)
/D1 24,D2 48,D3 72,D4 96/
RTU(J) RELEASE TIME OF UNIT J
/J1 0,J2 0,J3 0/
```

```
SCALAR
M large positive number/100/
NU Max no. of batches/5/
NL Min no. of batches/1/

VARIABLES
Y(I,J,K)        assignment of order I to slot K
TE(J,K)         start time of order I(day)
DE(I,D)         delay in delivery of order I(days)
H               makespan
OBJ             objective
B(I,J,K)        batch size
N(I,J,K)        number of batches;
POSITIVE VARIABLES B,TE,DE,XX;
BINARY VARIABLE Y;
INTEGER VARIABLE N;
EQUATIONS
PROD4b(I1,J,K)       Timing constraint
DELAY(I,J,K,D)       Delay constraint
EX1(J,K)             Assignment
EX4(I,J,K,D)         Extra slots
EXX1(I,J,K)          Transition1
EXX2(I,J,K)          Transition2
RT(J)                Release times
BATCH1(I,J,K,D)      Batch limit 1
BATCH2(I,J,K)        Batch limit 2
BATCH3(I,J,K,D)      Batch limit 3
BATCH4(I,J,K)        Batch limit 4
PROD2(I,D)           Production
OBJECTIVE            Minimize tardiness or makespan or tardy orders;

PROD4b(I1,J,K)$(JI1(J,I1) and (ord(K) GT 1)).. TE(J,K)=G=TE(J,K-1)+
SUM(I$IJ(I,J),XX(I,I1,J,K)*CT(I1,I))+N(I1,J,K)*FPT(J,I1)+B(I1,J,K)*PT(J,I1);

DELAY(I,J,K,D)$(JI(J,I) and KD(K,D)).. DE(I,D)=G=TE(J,K)-DD(D)-M*(1-Y(I,J,K));

RT(J).. TE(J,"K0")=E=RTU(J);

EX1(J,K).. SUM(I$IJ(I,J),Y(I,J,K))=E=1;

EXX1(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K-1)=E=sum(I1$I1J(I1,J),XX(I,I1,J,K));

EXX2(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K)=E=sum(I1$I1J(I1,J),XX(I1,I,J,K));

EX4(I,J,K,D)$((ORD(K) GT 2) AND LKD(K,D)).. XX(I,I,J,K+1)=G=XX(I,I,J,K);
```

```
PROD2(I,D)$ID(I,D)..
SUM(J$JI(J,I),SUM(K$KD(K,D),B(I,J,K)))=G=
SUM(D1$(ID1(I,D1) AND (ord(D1) LE ord(D))),Q(I,D1));

BATCH1(I,J,K,D)$(IJ(I,J) AND ORD(K) GT 1).. N(I,J,K)=L=NU*(Y(I,J,K)-
XX(I,I,J,K)$(NOT FIRSTK(K)));

BATCH2(I,J,K)$(IJ(I,J) AND ORD(K) GT 1).. B(I,J,K)=G=BSMIN(J,I)*N(I,J,K);

BATCH3(I,J,K,D)$(IJ(I,J) and ORD(K) GT 1).. N(I,J,K)=L=NU*Y(I,J,K);

BATCH4(I,J,K)$(IJ(I,J) and ORD(K) GT 1).. B(I,J,K)=L=BSMAX(J,I)*N(I,J,K);

OBJECTIVE.. OBJ=E=SUM(D,SUM(I,ALPHA(I,D)*DE(I,D)));

MODEL BATCH /all/;
BATCH.iterlim=100000000;
BATCH.reslim=100000000;
Y.UP(I,J,K)=1;
XX.UP(I,I1,J,K)=1;
Y.FX("I4","J1","K0")=1;
Y.FX("I1","J2","K0")=1;
Y.FX("I2","J3","K0")=1;
Y.FX("I2","J1","K0")=0;
Y.FX("I3","J1","K0")=0;
Y.FX("I3","J2","K0")=0;
Y.FX("I4","J2","K0")=0;
Y.FX("I1","J3","K0")=0;
N.UP(I,J,K)=5;
N.LO(I,J,K)=0;
B.LO(I,J,K)=0;
*option optca = .01;
option optcr = 0.0001;
OPTION SOLPRINT=OFF;
SOLVE BATCH USING MIP MINIMIZING OBJ;

DISPLAY TE.L,Y.L,XX.L,DE.L,OBJ.L,N.L,B.L;
```

B.2: GAMS file for Example 2 for Model M2

```
SETS
I         products /I1*I4/
I1(I)     products /I1*I4/
I2(I1)    products /I1*I4/
I3(I1)    products /I1*I4/
D         duedates /D1*D4/
D1(D)     duedates /D1*D4/
J         units    /J1*J3/
J1(J)     units    /J1*J3/
K         slots    /K0*K5/
KA(K)     slots    /K0*K5/
FIRSTK(K) first slots /K1,K2,K3,K4/

IJ(I,J)          products that can be produced on line J
/(I2*I4).J1,(I1,I3*I4).J2,(I1*I2).J3/
I1J(I1,J)        products that can be produced on line J
/(I2*I4).J1,(I1,I3*I4).J2,(I1*I2).J3/
I2J(I2,J)        products that can be produced on line J
/(I2*I4).J1,(I1,I3*I4).J2,(I1*I2).J3/
ID(I,D)          products due on D
/(I1*I3).D1,(I1*I4).D2,(I1*I4).D3,(I1*I2,I4).D4/
ID1(I,D1)        products due on D
/(I1*I3).D1,(I1*I4).D2,(I1*I4).D3,(I1*I2,I4).D4/
JI(J,I)          lines that can produce product I
/(J2*J3).I1,(J1,J3).I2,(J1*J2).I3,(J1*J2).I4/
JI1(J,I1)        lines that can produce product I
/(J2*J3).I1,(J1,J3).I2,(J1*J2).I3,(J1*J2).I4/
KD(K,D)          slots to take place before duedate D
/(K1).D1,(K1*K2).D2,(K1*K3).D3,(K1*K5).D4/
LKD(K,D)         slots to take place except last slot before duedate D
/(K1).D1,(K2).D2,(K3).D3,(K4).D4/

TABLE BSMIN(J,I) Min. Batch size of I on J
     I1   I2   I3   I4
J1        100  100  150
J2   100       100  100
J3   140  80

TABLE BSMAX(J,I) Max. Batch size of I on J
     I1   I2   I3   I4
J1        140  150  200
J2   120       120  150
J3   160  120
```

```
TABLE FPT(J,I) Fixed Processing Time of I on J
    I1   I2   I3   I4
J1       3    2    2.5
J2  5         2    3
J3  5    4

TABLE PT(J,I) Variables Processing Time of I on J
      I1    I2     I3    I4
J1          0.15   .17   0.15
J2  .15            .17   0.155
J3  .145  .155

TABLE CT(I1,I) Transition time bet I1&I on J
      I1   I2   I3   I4
I1  0.0  1.5  1.6  2.7
I2  5.1  0.0  1.3  4.8
I3  1.6  2.3  0.0  1.4
I4  1.0  2.5  2.1  0.0

TABLE Q(I,D) Amount of product I required at duedate D
     D1   D2    D3    D4
I1  50    100   100   100
I2  100   100   100   200
I3  50    100   100
I4        200   100   100

TABLE ALPHA(I,D) Weight coefficient for delay of product I at due date D
    D1   D2   D3   D4
I1  1    1    1    1
I2  1    1    1    1
I3  1    1    1    1
I4  1    1    1    1

PARAMETER
DD(D) due date D(hr)
/D1 24,D2 48,D3 72,D4 96/
RTU(J) RELEASE TIME OF UNIT J
/J1 0,J2 0,J3 0/

SCALAR
M large positive number/100/
NU Max no. of batches/5/
NL Min no. of batches/1/

VARIABLES
Y(I,J,K)         assignment of order I to slot K
```

```
TE(J,K)         start time of order I(day)
DE(I,D)         delay in delivery of order I(days)
H               makespan
OBJ             objective
DB(I,J,K)       batch size
DN(I,J,K)       number of batches;
POSITIVE VARIABLES TE,DE,XX,DB;
BINARY VARIABLE Y;
INTEGER VARIABLE DN;
EQUATIONS
PROD4b(I1,J,K)      Timing constraint
DELAY(I,J,K,D)      Delay order constraint
EX1(J,K)            Assignment
EX4(I,J,K,D)        Extra slots
EXX1(I,J,K)         Transition
EXX2(I,J,K)         Transition
RT(J)               Release time
BATCH1(I,J,K)       No. of batches limit
BATCH2(I,J,K)       Batch amount
PROD2(I,D)          Production
OBJECTIVE           Minimize tardiness or makespan or tardy orders;

PROD4b(I1,J,K)$(JI1(J,I1) and (ord(K) GT 1)).. TE(J,K)=G=TE(J,K-1)+
SUM(I$IJ(I,J),(XX(I,I1,J,K))*CT(I1,I))+(NL*(Y(I1,J,K)-XX(I1,I1,J,K)$(NOT
FIRSTK(K)))+DN(I1,J,K))*FPT(J,I1)
+((NL*(Y(I1,J,K)-XX(I1,I1,J,K)$(NOT
FIRSTK(K)))+DN(I1,J,K))*BSMIN(J,I1)+DB(I1,J,K))*PT(J,I1);

DELAY(I,J,K,D)$(JI(J,I) and KD(K,D)).. DE(I,D)=G=TE(J,K)-DD(D)-M*(1-Y(I,J,K));

RT(J).. TE(J,"K0")=E=RTU(J);

EX1(J,K).. SUM(I$IJ(I,J),Y(I,J,K))=E=1;

EXX1(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K-1)=E=sum(I1$I1J(I1,J),XX(I,I1,J,K));

EXX2(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K)=E=sum(I1$I1J(I1,J),XX(I1,I,J,K));

EX4(I,J,K,D)$(LKD(K,D) and ord(k) gt 2).. XX(I,I,J,K+1)=G=XX(I,I,J,K);

PROD2(I,D)$(ID(I,D))..
SUM(J$JI(J,I),SUM(K$KD(K,D),((NL*(Y(I,J,K)-XX(I,I,J,K)$(NOT
FIRSTK(K)))+DN(I,J,K))*BSMIN(J,I)+DB(I,J,K))))
=G=SUM(D1$(ID1(I,D1) AND (ord(D1) LE ord(D))),Q(I,D1));
```

```
BATCH1(I,J,K)$(IJ(I,J) and (ORD(K) GT 1)).. DN(I,J,K)=L=(NU-NL)*(Y(I,J,K)-
XX(I,I,J,K)$(NOT FIRSTK(K)));

BATCH2(I,J,K)$(IJ(I,J) AND (ORD(K) GT 1)).. DB(I,J,K)=L=(NL*(Y(I,J,K)-
XX(I,I,J,K)$(NOT FIRSTK(K)))+DN(I,J,K))*(BSMAX(J,I)-BSMIN(J,I));

OBJECTIVE.. OBJ=E=SUM(D,SUM(I,ALPHA(I,D)*DE(I,D)));


MODEL BATCH /all/;
BATCH.iterlim=100000000;
BATCH.reslim=100000000;
Y.UP(I,J,K)=1;
XX.UP(I,I1,J,K)=1;
DN.LO(I,J,K)=0;
DB.LO(I,J,K)=0;
Y.FX("I4","J1","K0")=1;
Y.FX("I1","J2","K0")=1;
Y.FX("I2","J3","K0")=1;
Y.FX("I2","J1","K0")=0;
Y.FX("I3","J1","K0")=0;
Y.FX("I3","J2","K0")=0;
Y.FX("I4","J2","K0")=0;
Y.FX("I1","J3","K0")=0;
*option optca = .01;
option solprint=off;
option optcr = 0.0001;
SOLVE BATCH USING MIP MINIMIZING OBJ;
PARAMETER AMT(I,J,K);
AMT(I,J,K)$(IJ(I,J) AND ORD(K) GT 1)=
(NL*(Y.L(I,J,K)-XX.L(I,I,J,K)$(NOT
FIRSTK(K)))+DN.L(I,J,K))*BSMIN(J,I)+DB.L(I,J,K);

DISPLAY TE.L,Y.L,XX.L,DE.L,OBJ.L,DN.L,DB.L,AMT;
```

APPENDIX C: Chapter 6

C.1: GAMS file

```
option solprint=off;

SETS
I Layers/I1*I8/
I1      /I1*I8/
I2      /I1*I8/
I3      /I1*I8/
I4      /I1*I8/
J       Steppers/J1*J4/
K       Slots/K0*K12/
L       Scenarios /1*2/
D       Shifts or Duedates/D0*D4/
D1(D) /D0*D4/
KA(K) slots /K0*K12/
FIRSTK(K) first slots/K1,K4,K7,K10/


IJ(I,J)         products that can be produced on line J
/(I1,I5,I7).J1,(I1,I6,I8).J2,(I1*I3,I7).J3,(I3*I4,I6).J4/
I1J(I1,J)       products that can be produced on line J
/(I1,I5,I7).J1,(I1,I6,I8).J2,(I1*I3,I7).J3,(I3*I4,I6).J4/
I2J(I2,J)       products that can be produced on line J
/(I1,I5,I7).J1,(I1,I6,I8).J2,(I1*I3,I7).J3,(I3*I4,I6).J4/
I3J(I3,J)       products that can be produced on line J
/(I1,I5,I7).J1,(I1,I6,I8).J2,(I1*I3,I7).J3,(I3*I4,I6).J4/
I4J(I4,J)       products that can be produced on line J
/(I1,I5,I7).J1,(I1,I6,I8).J2,(I1*I3,I7).J3,(I3*I4,I6).J4/
JI(J,I)         lines that can produce product I
/(J1*J3).I1,(J3).I2,(J3*J4).I3,(J4).I4,(J1).I5,(J2,J4).I6,(J1,J3).I7,(J2).I8/
JI1(J,I1)       lines that can produce product I
/(J1*J3).I1,(J3).I2,(J3*J4).I3,(J4).I4,(J1).I5,(J2,J4).I6,(J1,J3).I7,(J2).I8/
KD(K,D)         slots to take place before duedate D
/(K1*K3).D1,(K1*K6).D2,(K1*K9).D3,(K1*K12).D4/
LKD(K,D)        slots to take place except last slot before duedate D
/(K1*K2).D1,(K4*K5).D2,(K7*K8).D3,(K10*K11).D4/
SKD(K,D)        slots that are in between d and (d-1)
/(K1*K3).D1,(K4*K6).D2,(K7*K9).D3,(K10*K12).D4/


TABLE ST(J,L,I) SETUP TIME OF I ON J
     I1  I2  I3  I4  I5  I6  I7  I8
J1.1  .5          .7      .6
```

```
J2.1  .5                .6      .65
J3.1  .55 .6  .65             .65
J4.1        .6  .55     .6
J1.2  1.1           1       .9
J2.2  1                 1       1.1
J3.2  .9  1   1.1           1
J4.2        .95 1       1


TABLE PT(J,L,I) PROCESSING TIME OF I ON J
       I1     I2      I3      I4    I5      I6      I7      I8
J1.1  .7                      .8             .7
J2.1  .85                              .95           1
J3.1  .85    .9      .95                         1
J4.1               1.1     .75          .85
J1.2  .9                    1.2           1
J2.2  1.1                             1             1
J3.2  1.2    1       1.1                         .8
J4.2              1       .9          1.1

PARAMETER
DD(D) due date D(hr)
/D1 24,D2 48,D3 72,D4 96/
RTU(J) RELEASE TIME OF UNIT J
/J1 0,J2 0,J3 0/
BS(I)
/I1 25,I2 25,I3 25,I4 25,I5 25,I6 25,I7 25,I8 25/
P(I)
/I1*I8 300/
W(L)
/1 0.5,2 0.5/
WT(I)
/I1 1,I2 1.25,I3 1.75,I4 2,I5 1,I6 1.25,I7 1.75,I8 2/
SCALAR
M large positive number/1000/
NU Max no. of batches/200/
NL Min no. of batches/1/
H  HORIZON /96/

VARIABLES
Y(I,J,K)        assignment of order I to slot K
TE(J,K,L)       start time of order I(day)
*DE(I,D)        delay in delivery of order I(days)
*ST(I,D)        storage of layer I at due date D
PRO             production
B(I,J,K,L)      batch size
```

```
N(I,J,K,L)       number of batches;
POSITIVE VARIABLES SI,TE,XX,B;
BINARY VARIABLE Y;
INTEGER VARIABLE N;
EQUATIONS
PROD4b(J,K,L)        Timing constraint
EX1(J,K)             Assignment constraints
EX4(I,J,K,D)         Extra slots
EXX1(I,J,K)          Transition1
EXX2(I,J,K)          Transition2
RT(J,L)              Release time
BATCH1(I,J,K,L)      Batch limit 1
BATCH2(I,J,K,L)      Batch limit 2
BATCH3(I,J,K,L)      Batch limit 3
STORAGE1(I,D,L)      Storage 1
STORAGE2(I,D,L)      Storage 2
TIMELIMIT(J,K,L,D)
OBJECTIVE2     MAXIMIZING PRODUCTION;


RT(J,L).. TE(J,"K0",L)=E=RTU(J);

EX1(J,K).. SUM(I$IJ(I,J),Y(I,J,K))=E=1;

EXX1(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K-1)=E=sum(I1$I1J(I1,J),XX(I,I1,J,K));

EXX2(I,J,K)$(IJ(I,J) and ord(k) gt 1).. Y(I,J,K)=E=sum(I1$I1J(I1,J),XX(I1,I,J,K));

EX4(I,J,K,D)$(LKD(K,D) and ord(k) gt 2).. XX(I,I,J,K+1)=G=XX(I,I,J,K);

PROD4b(J,K,L)$((ord(K) GT 1)).. TE(J,K,L)=G=TE(J,K-1,L)
+SUM(I$IJ(I,J),(Y(I,J,K)-
XX(I,I,J,K))*ST(J,L,I))+SUM(I$IJ(I,J),(N(I,J,K,L)*PT(J,L,I)));

BATCH1(I,J,K,L)$(IJ(I,J) AND ORD(K) GT 1).. N(I,J,K,L)=L=NU*(Y(I,J,K)-
XX(I,I,J,K)$(NOT FIRSTK(K)));

BATCH2(I,J,K,L)$(IJ(I,J) AND ORD(K) GT 1).. B(I,J,K,L)=E=BS(I)*N(I,J,K,L);

BATCH3(I,J,K,L)$(IJ(I,J) and ORD(K) GT 1).. N(I,J,K,L)=L=NU*Y(I,J,K);

STORAGE1(I,D,L)$(ORD(D) GT 1).. SI(I,D,L)=E=SI(I,D-1,L)+P(I)-
SUM(J,SUM(K$SKD(K,D),B(I,J,K,L)));

STORAGE2(I,D,L)$(ORD(D) GT 1).. SI(I,D-
1,L)=G=SUM(J,SUM(K$SKD(K,D),B(I,J,K,L)));
```

```
TIMELIMIT(J,K,L,D)$(KD(K,D)).. TE(J,K,L)=L=DD(D);

OBJECTIVE2.. PRO=E=SUM((I,J,K,L)$(IJ(I,J) AND ORD(K) GT
1),B(I,J,K,L)*W(L)*WT(I));

MODEL BATCH /all/;
BATCH.iterlim=100000000;
BATCH.reslim=100000000;
Y.UP(I,J,K)=1;
XX.UP(I,I1,J,K)=1;
N.LO(I,J,K,L)=0;
B.LO(I,J,K,L)=0;
Y.FX("I5","J1","K0")=1;
Y.FX("I8","J2","K0")=1;
Y.FX("I2","J3","K0")=1;
Y.FX("I6","J4","K0")=1;
Y.FX("I1","J1","K0")=0;
Y.FX("I7","J1","K0")=0;
Y.FX("I1","J2","K0")=0;
Y.FX("I6","J2","K0")=0;
Y.FX("I1","J3","K0")=0;
Y.FX("I3","J3","K0")=0;
Y.FX("I7","J3","K0")=0;
Y.FX("I3","J4","K0")=0;
Y.FX("I4","J4","K0")=0;
SI.FX("I1","D0",L)=400;
SI.FX("I2","D0",L)=400;
SI.FX("I3","D0",L)=400;
SI.FX("I4","D0",L)=400;
SI.FX("I5","D0",L)=400;
SI.FX("I6","D0",L)=400;
SI.FX("I7","D0",L)=400;
SI.FX("I8","D0",L)=400;
*option optca = .01;
option optcr = 0.01;
OPTION SOLPRINT=OFF;
SOLVE BATCH USING MIP MAXIMIZING PRO;
PARAMETER AMT(L);
AMT(L)=SUM((I,J,K)$(IJ(I,J) AND ORD(K) GT 1),B.L(I,J,K,L));

DISPLAY TE.L,Y.L,PRO.L,N.L,B.L,SI.L,XX.L,AMT;
```