SHORT TERM SCHEDULING OF BATCH & CONTINUOUS PROCESSES

By
SHOBIT
R670209025

College of Engineering
University of Petroleum & Energy Studies
Dehradun
May, 2011

SHORT TERM SCHEDULING OF BATCH & CONTINUOUS PROCESSES

A thesis submitted in partial fulfillment of the requirements for the Degree of
Master of Technology
(Process Design Engineering)

By
SHOBIT

Under the guidance of

Mr. ADARSH ARYA
Asst. Professor
College of Engineering Studies
University of Petroleum & Energy Studies

Approved

.....
Dean
College of Engineering
University of Petroleum & Energy Studies
Dehradun
May, 2011

CERTIFICATE

This is to certify that the work contained in this thesis titled **"SHORT TERM SCHEDULING OF BATCH & CONTINUOUS PROCESSES"** has been carried out by **SHOBIT** under my/our supervision and has not been submitted elsewhere for a degree.

.....

Mr Adarsh Arya
Asst. Professor
College of Engineering studies
University of Petroleum & Energy studies

05/05/11
.....
Date

ACKNOWLEDGEMENT

I owe a deep sense of gratitude to my project guide Asst. Prof. Adarsh Arya, COES, UPES for providing me the opportunity to work on this project, providing the required resources and for his able guidance and constant encouragement which helped me in successful completion of my work.

I owe a deep sense of gratitude to Course co-ordinator Prof. G. Sanjay Kumar, COES, UPES for his able guidance and constant encouragement which helped me in successful completion of my work.

I owe a deep sense of gratitude to my external project guide Dr. Munnawar Abdul Sheikh, Asst Professor ,Deptt Of chemical engineering , IIT Delhi for providing me the opportunity to work on this project, providing the required resources and for his able guidance and constant encouragement which helped me in successful completion of my work.

I owe a deep sense of gratitude to my project mate Mr. Sanjeev Yadav and Mr. Ramasagar who helped me each day of project work.

I thank my family and friends who directly or indirectly helped in completion of the Project.

Shobit

Abstract

Operations scheduling of chemical process industries has been great matter of attention in last couple of decades because of its importance in increasing the capacity of plant performance and reducing the cost associated with running the plant. Numerous formulations have been proposed to address the problems of optimizing short term and cyclic schedules of batch and continuous plants based on different time and event based model.. Short term scheduling of batch and continuous processes is a critical issue to ensure optimal use of resources, such as raw materials or equipments in process plants. The objectives of the process may either be to maximize profit or to minimize cost of production or make span. This requires best possible allocation of batches to the time horizon of interest. This can be done either by discrete time formulation or continuous time formulation. Discrete time formulation involves the division of time into equal intervals over the time horizon such that the starting or ending of a task can take place only at an event point. Continuous time formulations involve allocating the starting or ending of the task at any point in the time horizon. Continuous time formulations have been found to be more advantageous as compared to discrete time in terms of model complexity, number of variables in the model and the computation time.

Oil refineries are increasingly concerned with improving the planning of their operations and optimizing not only single production units but the whole production enterprise. However, the modeling of the overall refinery operation from the crude oil arrival to the distribution of oil products gives rise to intractable mathematical models. Researchers in the past have developed decomposition technique to reduce the computational burden. An effort has been made to understand short term scheduling of Refinery operation and to analyze benchmark examples from literature and draw conclusion on analyses.

When a plant in operation encounters an unforeseen event either in the form of machine breakdown or addition/ deletion of customer orders, reactive scheduling comes into picture. Reactive scheduling requires information about the original schedule before the

occurrence of the deviation and the time and nature of deviation. It then creates a schedule which takes into account the deviation, keeping in view the objective of the process.

As a prerequisite to reactive scheduling, an effort has been made to understand short term scheduling and to implement benchmark examples from literature.

CONTENTS

List of Figures

List of Tables

Chapter 1
Introduction

1.1 General Scheduling

Optimal use of the resources for getting the maximum profit has always been the area of great concern for the people working in industry and in academics. Scheduling in process operations refers to the procedures of allocating the different resources (processing units, materials, utilities etc) to the processing tasks required to manufacture one or more product and is crucial for improving production performance. Significant amount of work has appeared in chemical literature concerning the scheduling. Reklaitis (1992) reviewed the scheduling and planning of batch process operations, focusing on the basic elements of scheduling problems of chemical manufacturing systems and the available solution methods.

Rippin (1993) summarized the development of batch process systems engineering with particular reference to the areas of design, planning, scheduling and uncertainty. Shah (1998) examined different techniques for optimizing production schedules. Pekny and Reklaitis (1998) discussed the nature and characteristics of the scheduling/planning and pointed out the key implications for the solution methodology for these problems. Pinto and Grossmann (1998) presented an overview of assignment and sequencing models used in process scheduling with mathematical programming techniques. Recently Mendez at al (2006) provided an up-to-date review of the state-of-the-art in this challenging area.

There are a great variety of aspects that need to be considered when developing scheduling models for batch and continuous processes. Types of process, process topology, inventory policies, demand patterns, resource constraints have great influence on formulation. Time horizon, Changeovers, Batch size, batch processing times, Costs associated with equipments, utilities, Inventories also put additional complexities to the model. This diversity of factors makes the task of developing unified general methods

quite difficult. At the same time, there might be the trade-off of having a number of specialized methods that can address specific cases of this classification in a more efficient way.

A roadmap is introduced that describes the main features of current optimization approaches. This section is of particular importance because alternative ways of addressing/formulating the same problem. These usually have a direct impact on the computational performance, capabilities and limitations of the resulting optimization model.

1.1.1 Time representation

First and most important issue is the time representation. Depending on whether the events of the schedule can only take place at some predefined time points, or can occur at any moment during the time horizon of interest, optimization approaches can be classified into discrete and continuous time formulations. Discrete time models are based on: (i) dividing the scheduling horizon into a finite number of time intervals with predefined duration and, (ii) allowing the events such as the beginning or ending of tasks to happen only at the boundaries of these time periods. Therefore, scheduling constraints have only to be monitored at specific and known time points, which reduces the problem complexity and makes the model structure simpler and easier to solve.

In continuous time formulations, timing decisions are explicitly represented as a set of continuous variables defining the exact times at which the events take place. In the general case, a variable time handling allows obtaining a significant reduction of the number of variables of the model and at the same time, more flexible solutions in terms of time can be generate

1.1.2 Material Balance

The handling of batches and batch sizes gives rise to two types of optimization model categories. The first category refers to monolithic approaches, which simultaneously deal with the optimal set of batches (number and size), the allocation and sequencing of manufacturing resources and the timing of processing tasks. These methods are able to deal with arbitrary network processes involving complex product recipes. These models employ the state-task network (STN), or the resource-task network (RTN) concept to represent the problem. As shown in Fig. 1, the STN-based models represent the problem assuming that processing tasks produce and consume states (materials). A special treatment is given to manufacturing resources aside from equipment. The STN is a directed graph that consists of three key elements: (i) *state nodes* representing feeds, intermediates and final products; (ii) *task nodes* representing the process operations which transform material from one or more input states into one or more output states and; (iii) *arcs* that link states and tasks indicating the flow of materials. State and task nodes are denoted by circles and rectangles, respectively.

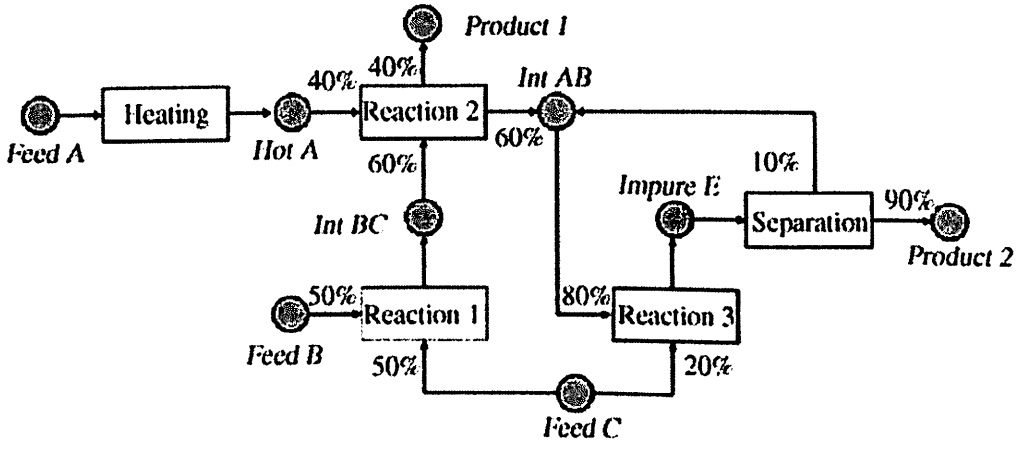

Figure 1.1: STN Representation

In contrast, the RTN-based formulations employ a uniform treatment and representation framework for all available resources through the idea that processing and storage tasks consume and release resources at their beginning and ending times (see Fig. 2). In this particular case, circles represent not only states but also other resources required in the batch process such as processing units and vessels.

The second category comprises models that assume that the number of batches of each size is known in advance which decomposes the whole problem into two stages, batching and batch scheduling. The batching problem converts the primary requirements of products into individual batches aiming at optimizing some criterion like the plant workload. Afterwards, the available manufacturing resources are allocated to the batches over time

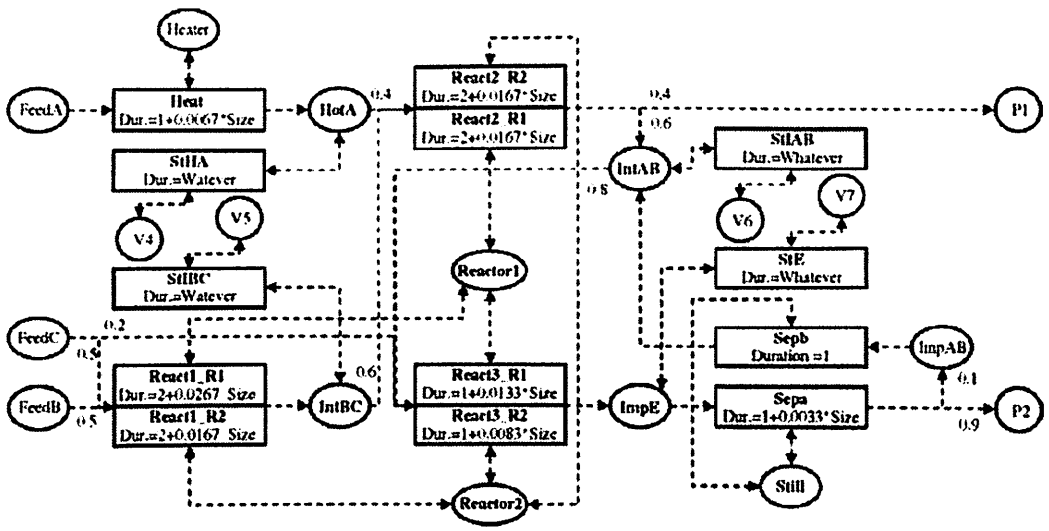

Figure 1.2: RTN Representation

1.1.3 Event Representation

In addition to the time representation and material balances, scheduling models are based on different concepts or basic ideas that arrange the events of the schedule over time with the main purpose of guaranteeing that the maximum capacity of the shared resources is never exceeded.

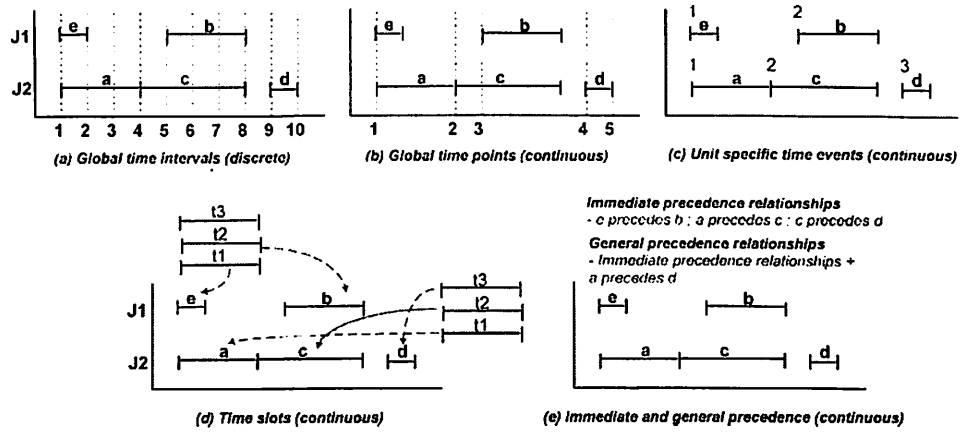

Figure 1.3: Different concepts of event representation

As can be seen in Fig. 3, we classified these concepts into five different types of event representations, which have been broadly utilized to develop a variety of mathematical formulations for batch scheduling problems. Particularly, Fig. 3 depicts a schematic representation of the same schedule obtained by using the alternative concepts. The small example given involves five batches (a, b, c, d, e) allocated to two units (J1, J2). To represent this solution, the different alternatives require: (a) 10 fixed time intervals, (b) five variable global time points, (c) three unit-specific time events, (d) three asynchronous time slots for each unit, (e) three immediate precedence relationships or four general precedence relationships.

For discrete time formulations, the definition of global time intervals is the only option for general network and sequential processes. In this case, a common and fixed time grid valid for all shared resources is predefined and batch tasks are enforced to begin and finish exactly at a point of the grid. In contrast to the discrete time representation, continuous time formulations involve extensive alternative event representations. The global time point representation corresponds to a generalization of global time intervals where the timing of time intervals is treated as new model variable. In this case, a common and variable time grid is defined for all shared resources. The beginning and the finishing times of the set of batch tasks are linked to specific time points through the key

discrete variables. Models following this direction are relatively simple to implement even for general scheduling problems.

In contrast to global time points, the idea of unit-specific time events defines a different variable time grid for each shared resource, allowing different tasks to start at different moments for the same event point. Because of the heterogeneous locations of the event points, the number of events required is usually smaller than in the case of global time points.

Slot based continuous-time formulations were initially focused on a wide variety of sequential processes, have been recently extended to also consider general batch processes. Time slots stands for a set of predefined time intervals with unknown durations. The main idea is to postulate an appropriate number of time slots for each processing unit in order to allocate them to the batch tasks to be performed. The selection of the number of time slots required is not a trivial decision and represents an important trade-off between optimality and computational performance. Slot-based representations can be classified into two types: synchronous and asynchronous. The synchronous representation, which is similar to the idea of global time points, defines identical or common slots across all units in such a way that the shared resources involved in network batch processes are more natural and easier to handle. Alternatively, the asynchronous representation allows the postulated slots to differ from one unit to another, which for a given number of slots provides more flexibility in terms of timing decisions than its synchronous counterpart.

1.1.4 Objective function:

There are different measures of the quality of the solution that can be used for scheduling problems. Typical measures of overall objective function in process scheduling problems include minimize makespan, minimize earliness, minimize tardiness, minimize costs, and maximize profit.

1.2 Objective

We aim to understand short term scheduling models and the objective of this project is to implement both continuous and discrete time formulations of benchmark examples of short term scheduling from literature.

There are few models available for the short-term scheduling of refinery problem but most of them rely on special assumptions that generally make the solution inefficient or unrealistic for real world cases. To account for the major weaknesses of the available mathematical approaches and suggest for improvement is the major goal of this project.

Chapter 2
Literature Review

A critical review is given on the scheduling of different chemical processes. There has been significant progress in the area of short-term scheduling of chemical processes, including the solution of industrial-sized problems, in the last 20 years. Main features, strengths and limitations of existing modeling and optimization techniques as well as other available major solution methods are reviewed. Literature on the scheduling of chemical processes can be divided into several classes on the basis of time and event representation.

2.1 General Scheduling

2.1.1 Discrete time representation (Global time intervals)

The event representation based on the definition of global time intervals employs a predefined time grid that is valid for all shared resources involved in the scheduling problem, such as processing units. Events such as the beginning and ending of a task are associated with the boundaries of these time intervals. Two of the earliest research contributions that employed this type of discrete time representation were presented by Bowman (1959) and Manne (1960) for job shop scheduling problems in the operations research community literature. There have been notable subsequent developments, for example, those by Pritsker, Watters, and Wolfe (1969) for resource-limited multiproject and job shop scheduling.

Relevant modeling features of discrete models are based on the STN and RTN process representation. The most relevant contribution for discrete time models is the state task network representation proposed by Kondili, Pantelides, and Sargent (1993) and Shah, Pantelides, and Sargent (1993) (see also Rodrigues, Latre,& Rodrigues, 2000). The STN model covers all the features that are included at the column on discrete time in Table 1. A simpler and general discrete time scheduling formulation can also be derived by means of the resource task network concept proposed by Pantelides (1994). The major advantage of the RTN formulation over the STN counterpart arises in problems involving

identical equipment. Here, the RTN formulation introduces a single binary variable instead of the multiple variables used by the STN model. The RTN-based model also covers all the features at the column on discrete time in Table 1. The main limitations of the time discretization methods are that (i) they correspond to an approximation of the time horizon and (ii) they result in an unnecessary increase of the number of binary variables in particular and of the overall size of the mathematical model. As a result of these shortcomings of the time discretization methods, recent work aims at developing efficient methods based on the continuous time representation.

2.1.2. Continuous time representation

Due to the inherent limitations of the discrete-time approaches, there has been a significant amount of attention on the development of continuous-time representations in the past decade. We classify all continuous-time approaches into two categories based on the type of processes. The first category of approaches focuses on sequential processes and the second category aims at the scheduling of general network-represented processes. The critical differences between these two types of processes is that sequential processes are order or batch oriented and do not require the explicit consideration of mass balances, which has important implications for the modeling of related scheduling problems.

Sequential processes

One of the first approaches to formulate continuous-time models for the scheduling of sequential processes, which can be single or multi-stage, is based on the concept of time slots. At each stage, there can be one or multiple parallel units. When multiple units are involved, time slots are defined for each unit. The basic idea is illustrated in Fig. 5(b). Research contributions following this direction include those presented by Pinto and Grossmann (1994, 1995, 1996), Pinto, Ttirkay, Bolio, and Grossmann (1998), Karimi and McDonald (1997), Lamba and Karimi (2002a,b), Bok and Park (1998), Moon and Hrymak (1999).

Because of the batch or order oriented characteristics of the sequential processes, it is

possible to define continuous variables directly to represent the timings of the batches without the use of time slots. This alternative direction has also been pursued to formulate continuous-time scheduling models for sequential processes, as reported in the work presented by Ku and Karimi (1988), Cerdá, Henning, and Grossmann (1997), Méndez et al. (2000b, 2001), Moon, Park, and Lee (1996), Hui, Gupta, and van der Meulen (2000), Hui and Gupta (2001), Orçun, Altinel, and Hortasu (2001), and Lee, Heo, Lee, and Lee (2002).

General network-represented processes

For general network-represented processes that allow batches to merge/split and thus require explicit consideration of mass balance, two types of approaches have been developed to build continuous-time scheduling formulations. The first approach introduces a set of events or time slots that are used for all tasks and all units. We denote the formulations applying this approach as "global event based models." The second approach defines event points on a unit basis, allowing tasks corresponding to the same event point but in different units to take place at different times. This is the most general and most rigorous representation and we denote it as "unit-specific event based models."

Global event based models

There have been an increasing number of research contributions on continuous-time formulations for scheduling of general processes. The earliest efforts were presented by Zhang and Sargent (1996, 1998), Zhang (1995), Mockus and Reklaitis (1997, 1999a, b), and Schilling and Pantelides (1996, 1999). Recent developments include the work presented by Castro, Barbosa-Póvoa, and Matos (2001), Majozi and Zhu (2001), Lee, Park, and Lee (2001), Burkard, Fortuna, and Hurkens (2002) and Wang and Guignard (2002).

Most of these formulations have been based on either the STN or RTN process representations. The basic idea of the continuous-time scheduling models based on global

events is to introduce continuous variables to determine the timings of events or variable time slots and use binary variables to assign important state changes of the system, for example, the start or end of a task, to these events or time slots. Zhang and Sargent (1996, 1998) and Zhang (1995) developed the first such continuous-time model based on both STN and RTN for mixed production facilities involving both batch and continuous processes.

Mockus and Reklaitis (1997, 1999a, b) (also see Mockus, Eddy, Mockus, Mockus, & Reklaitis, 1997, part V) proposed a similar approach based on the STN framework and applied it to a variety of scheduling problems for multiproduct/ multipurpose batch and continuous plants. Their continuous-time formulation, which is called Non-Uniform Discrete-Time Model (NUDTM), also leads to large scale MINLP problems. They can be transformed into MILP problems if the objective function is of simple form. When a more complicated objective is involved (for example, the maximization of overall profit which takes into account storage cost and utility cost), they proposed a modified outer approximation (Duran & Grossmann, 1986) or a Bayesian heuristic approach to solve the resulting nonconvex MINLP problems. Schilling and Pantelides (1996, 1999) proposed a continuous-time formulation based on the RTN framework.

Unit-specific event based models

In order to gain more flexibility in timing decisions without increasing the number of time points to be defined, an original concept of event points was introduced by Ierapetritou and Floudas (1998). Ierapetritou and Floudas (1998a, b), Ierapetritou, Hene, and Floudas (1999), Lin and Floudas (2001), Ierapetritou and Floudas (2001) proposed a novel continuous-time formulation for short-term scheduling of batch, semicontinuous, and continuous processes. This formulation introduces an original concept of event points, which are a sequence of time instances located along the time axis of a unit, each Fig. 5. Event points defined for each unit representing the beginning of a task or utilization of the unit. The basic idea is illustrated in Figure 5 The location of event points are different

for different units, allowing different tasks to start at different moments in different units for the same event point. The timings of tasks are then accounted for through special sequencing constraints, as will be discussed in detail below.

Because of the heterogeneous locations of the event points for different units as well as the definition of an event as only the starting of a task (compared to that in a global-event based model which considers the starting and the finishing of a task as two events), for the same scheduling problem, the number of event points required in this formulation is smaller than the number of events in the global event based models described in the previous section. This results in substantial reduction of the number of binary variables. Compared to the discrete-time models and most of other continuous-time models, this formulation leads to MILP models of smaller size mainly in terms of the number of binary variables, which consequently requires less computational effort for their solution.

Janak, Lin and Floudas (2004), extended the work of Ierapetritou and Floudas (1998) and Lin and Floudas(2001) to account for resource constraints, various storage policies (UIS, FIS, NIS, and ZW), variable batch sizes and processing times, batch mixing and splitting, and sequence-dependent changeover times. In that model, tasks are allowed to continue over several consecutive event points in order to accurately monitor the utilization of resources and the storage of states so that specified limits are enforced.

Although JLF (2004), is relevant for batch plants with resources, allows task to continue over several event points, and can handle mixed storage policies, but computationally it does not perform well for problems without resources. Shaikh and Floudas (2008) bridged these gaps in the literature and unifies both problem with and with out resources to form a novel short term scheduling model using three index binary and continuous variables that efficiently merges both the problems involving resource and no resource constraints into a unified ,generic common framework.

2.2 Cyclic Scheduling

Campaign mode of operation can be selected in batch plant operation when demands and operating conditions are relatively stable. In this case, plants focus on producing only a subset of products over a certain time period. In the same context, cyclic scheduling is developed to make the operation decisions easier and profitable. It establishes an operation schedule and makes it executed repeatedly.

Shah, Pantelides, and Sargent (1993) modified the formulation of Kondili et al. (1993) formulation and extended it to the periodic scheduling of batch plants using a discrete time representation. Schilling and Pantelides (1999) presented a periodic scheduling formulation which is based on their earlier work on continuous-time representation for short-term scheduling problem. More recently, Castro, Barbosa-Povoa, and Matos (2003) modified their short-term scheduling formulation to fit periodic scheduling requirement for an industrial application. The proposed model is based on the basic RTN representation and encounters the main limitation of the prohibitive model size.

In most of the early work presented in the area of single-campaign scheduling, the cycle time of a particular product was commonly defined as the average elapsed time for the production of a batch of that product. This definition is not suitable for schedules in which different batches of the same product may be of different sizes. Wellons and Reklaitis (1989) consider a number of batches of potentially different sizes following different paths through the plant equipment. In this case, a cycle involves the production of exactly one batch along each of the paths, and the cycle time is defined as the shortest period between the start of successive cycles. In the multiple product campaign case, a different cycle time is defined for each product. However, neither of the above cycle time definitions is appropriate for schedules of the generality considered here. First, because of the possible coupling of products through shared intermediates, it is not always meaningful to define different cycle times for different products. Also relating cycle time to the production of one or more batches is not particularly helpful since batch identity is not necessarily preserved.

Shah, Pantelides, and Sargent (1993) considered a cycle as a sequence of operations potentially involving the production of all desired products, and the utilization of all available resources. The cycle time is the shortest time interval at which these operations can be repeated - thus, there is only one cycle time for all products being considered.

Wu and Ierapetritou (2004) extended the work by Ierapetritou and Floudas (1998a, b) based on the STN representation, and develops the cyclic scheduling formulation with the inherited advantage of using few binary variables.

Although the emphasis has always been on developing the cyclic schedule of batch plants but quite a few works has also been done on developing the cyclic schedule of continuous plant. Sahinidis and Grossmann (1990) addressed the problem of cyclic multiproduct scheduling on continuous parallel product lines. This plant configuration is typically used in the manufacturing of specialty chemicals plant. Pinto and Grossmann(1994) presented a novel formulation for cyclic scheduling of multistage, multiproduct continuous plant which consists of stages involving one production line that are interconnected by storage tanks. Castro and Novais (2007) presented a new mixed integer nonlinear program MINLP model for the periodic scheduling of multistage, multiproduct continuous plant featuring equipment units in parallel that are subject to sequence dependent changeovers. The formulation was based on RTN based presentation.

2.3 Short term scheduling of refinery operations

In the literature, mathematical programming technologies have been extensively concerned with and developed in the area of long-term refinery planning, while short-term scheduling has received less attention. Fewer publications have been appeared to address the short-term scheduling of refinery operations. (Zhang & Zhu, 2000) proposed a decomposition approach, which decomposes the overall refinery model into a site level and a process level. Pinto and Moro (2000) presented planning and scheduling model. Luo and Rong (2007) presented hierarchal approach for short term scheduling in

refineries. This paper focuses on a hierarchical approach with two decision levels for short-term scheduling problems in refineries. The optimization model at the upper level and the heuristics and rules adopted in simulation system at the lower level are presented.

The overall oil-refinery system is decomposed into three parts. The first part involves the crude-oil unloading, mixing and inventory control, the second part consists of the production unit scheduling which includes both fractionation and reaction processes, and the third part depicts the finished product blending and shipping end of the refinery.

Shah (1996) applied mathematical programming techniques to crude-oil scheduling. However the models are prohibitively expensive due to the nature of discrete time representation. The problem of crude-oil unloading with inventory control is addressed by Lee, Pinto, Grossmann, and Park (1996) based on time discretization and by Jia and Ierapetritou (2003) using continuous time based model. Reddy, Karimi and Srinivasan (2004) presented the first complete continuous-time (MILP) formulation for the short-term scheduling of crude-oil unloading operations and later improved the formulation by adding Revised Reddy's Algorithm(RAA).

Furman, Jia and Ierapetritou (2007) proposed a generalized model to robustly hand the synchronization of time events with material balances than the previous models in literature by Jia and Ierapetritou (2003). Karuppiah, Furman and Grossmann (2008) proposed a new decomposition approach to the MINLP problem proposed by Furman, Jia and Ierapetritou (2007). Very recently, Saharidis and Ierapetritou (2009) and Saharidis and Dallery(2009) proposed a new approach which provides not only the optimal scheduling but also the optimal type of mixture preparation.

Jia and Ierapetritou (2004) developed a comprehensive mathematical programming model based on a continuous time formulation for the scheduling of oil-refinery operations and decomposed the overall problems into three domains: the crude-oil unloading and blending, the production unit operations, and the product blending and delivery. Gasoline blending is a crucial step in refinery operation as gasoline can yield 60–70% of a refinery's profit. The process involves mixing various stocks, which are the intermediate products from the refinery, along with some additives, such as antioxidants and corrosion inhibitors, to produce blends with certain qualities.

Chapter 3

Scheduling of Refinery Crude Oil Operations

Planning and scheduling of the flow of crude oil is a very important problem in petroleum refineries due to the potential realization of large cost savings and improved feeds. The overall oil-refinery system can be decomposed into three parts as depicted in Figure. 10. The first part involves the crude-oil unloading, mixing and inventory control, the second part consists of the production unit scheduling which includes both fractionation and reaction processes, and the third part, depicts the finished product blending and shipping end of the refinery.

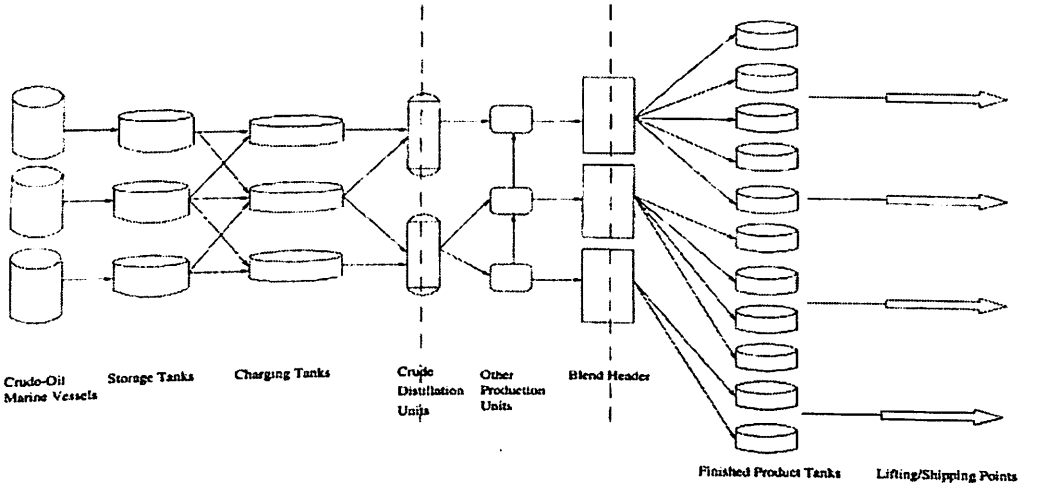

Figure 3.1: Graphical Overview of refinery system

Significant amount of work has been done on developing the schedules for the first problem. People have proposed different formulations using different time representations. Initial works were based on discrete time based formulation but proved prohibitive in nature when tested for large complex configurations of crude oil loading and unloading. But, in recent past, researchers have switched to continuous time based techniques to schedule the crude oil loading and unloading operations.

Jia and Ierapetritou (2003) addressed the problem of gasoline blending and distribution .The problem involves the optimal operation of gasoline blending, the transfer to product stock tanks and delivering schedule to satisfy all of the orders. An efficient mixed – integer linear programming formulation is developed based on continuous representation of time domain.

Summary and Future Directions

In this paper, a continuous-time formulation was presented for the short-term scheduling of a gasoline blending and distribution system. It is shown that the resulting model can be solved efficiently even for realistic large-scale problems. The main advantage of the proposed approach is the full utilization of the time continuity. This results in smaller models in terms of variables and constraints because only the real events have to be modeled. On the contrary, discrete-time formulations which are commonly used for refinery operations result in an excessive number of variables and constraints because of unnecessary time discretization. The relaxation of the fixed product recipe is our current focus. However, this results in a non convex formulation for which global optimization should be employed.

The overall refinery scheduling consists of all three stages: the scheduling for a crude oil charging system, the downstream production steps, and a finished product blending and shipping system. Previously published work addressed the crude oil unloading, mixing, and inventory control (Jia and Ierapetritou6). Work is currently performed that involves the production unit scheduling (Figure 1) as well as the integration of all three different problems (Figure 1) and will be the subject of future publications

Chapter 4

SHORT TERM SCHEDULING OF BATCH & CONTINUOUS PROCESSES

4.1 Introduction

The research area of batch and continuous process scheduling has received great attention from both the academia and the industry in the past two decades. This is motivated, on one hand, by the increasing pressure to improve efficiency and reduce costs, and on the other hand, by the significant advances in relevant modeling and solution techniques and the rapidly growing computational power.

In multiproduct and multipurpose batch, semicontinuous and continuous plants, different products are manufactured via the same or different sequence of operations by sharing available pieces of equipment, intermediate materials and other production resources. They have long been accepted for the manufacture of chemicals that are produced in small quantities and for which the production process or the demand pattern is likely to change. The inherent operational flexibility of these plants provides the platform for great savings reflected in good production schedules.

In general, scheduling is a decision making process to determine when, where and how to produce a set of products given requirements in a specific time horizon, a set of limited resources, and processing recipes. Due to the discrete decisions involved (e.g., equipment assignment, task allocation over time) these problems are inherently combinatorial in nature, and hence very challenging from the computational complexity point of view

Given the computational complexity of combinatorial problems arising from process scheduling, it is of crucial importance to develop effective mathematical formulations to model the manufacturing processes and to explore efficient solution approaches for such problems. All of the mathematical models in the literature can be classified into two main groups based on the time representations. Early attempts relied on the discretization of the time horizon into a number of time intervals and inevitably has the main

limitations of model inaccuracy (i.e., discrete approximation of the time horizon which leads to suboptimal solution by definition) and unnecessary increase of the overall size of the resulting mathematical programming problems due to the introduction of large number of binary variables associated with each discrete time interval. To address these limitations, methods based on continuous-time representations have attracted a great amount of attention and provide great potential for the development of more accurate and efficient modeling and solution approaches

4.2 Classification of scheduling formulations

4.2.1 Time representation

The key issue for process scheduling problems concerns the time representation. All existing scheduling formulations can be classified into two main categories: discrete-time models and continuous-time models. Early attempts in modeling the process scheduling problems relied on the discrete-time approach, in which the time horizon is divided into a number of time intervals of uniform durations and events such as the beginning and ending of a task are associated with the boundaries of these time intervals.

To achieve a suitable approximation of the original problem, it is usually needed to use a time interval that is sufficiently small, for example, the greatest common factor (GCF) of the processing times. This usually leads to very large combinatorial problems of intractable size, especially for real-world problems, and hence limits its applications.

The basic concept of the discrete-time approach is illustrated in Fig. 1 and further discussion.

Due to the aforementioned limitations of the discrete time approach, researchers have started developing continuous-time models in the past decade. In these models, events are potentially allowed to take place at any point in the continuous domain of time. Modeling of this flexibility is accomplished by introducing the concepts of variable event times, which can be defined globally or for each unit. Variables are required to determine the timings of events. The basic idea of the continuous-time approach is also illustrated in Fig. 1. Because of the possibility of eliminating a major fraction of the inactive event-time interval assignments with the continuous-time approach, the resulting mathematical programming problems are usually of much smaller sizes and require less computational efforts for their solution. However, due to the variable nature of the timings of the events,

it becomes more challenging to model the scheduling process and the continuous-time approach may lead to mathematical models with more complicated structures compared to their discrete-time counterparts.

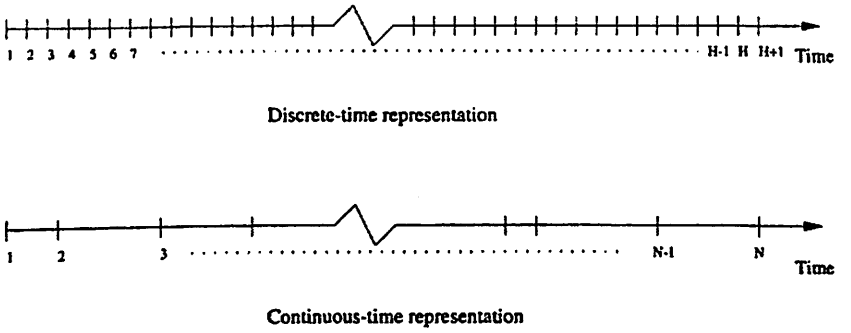

Fig. 1. Discrete and continuous representations of time.

Figure 4.1

4.2.2 Discrete-time approaches

In discrete-time approaches for scheduling problems, the time horizon of interest is divided into a number of time intervals of uniform durations. Events such as the beginning and ending of a task are associated with the boundaries of these time intervals. The main advantage of the discrete-time representation is that it provides a reference grid of time for all operations competing for shared resources, such as equipment items. This renders the possibility of formulating the various constraints in the scheduling problem in a relatively straightforward and simple manner,

Mathematical model

One of the common components in scheduling problems involves the allocation of units to tasks. To model these assignments, binary variables Wijt are introduced to determine whether or not a task (i) starts in unit (j) at the beginning of time interval (T) and the following allocation constraints are formulated:

$$\sum_{i \in I_j} W_{ijt} \leq 1, \quad \forall j \in J,\ t \in T. \tag{1}$$

$$\sum_{i' \in I_j} \sum_{t'=t}^{t+\alpha_{ij}-1} W_{i'jt'} - 1 \leq M(1 - W_{ijt}),$$
$$\forall j \in J,\ i \in I_j,\ t \in T. \tag{2}$$

where Ij is the set of tasks that can be performed in unit (j), αij is the fixed processing time of task (i) in unit (j) and M is a sufficiently large positive number. Constraint (1)states that at most one task can start in any unit at any time interval. Constraint (2) further expresses the requirement that if task (i) starts in unit (j) at time interval (T) (i.e., Wijt =1), then no other task can start in the same unit until task (i) is finished after the duration of αij . Note that the latter constraint becomes trivially satisfied when task (i) does not start at time (T) (i.e., Wijt = 0). To account for batch-sizes and mass balances, continuous variables Bijt are used to represent the amount of material which starts undergoing task (i) in unit (j) at time interval (T), and Sst is the amount of material state (s) during time interval (T). The following constraints are introduced to represent the relations among them and the corresponding binary variables

$$W_{ijt} V_{ij}^{\min} \leq B_{ijt} \leq W_{ijt} V_{ij}^{\min}, \quad \forall i \in I,\ j \in J_i,\ t \in T, \tag{3}$$

$$S_{st} = S_{s,(t-1)} + \sum_{i \in I_s^p} \rho_{is}^p \sum_{j \in J_i} B_{i,j,(t-\alpha_{is})} - \sum_{i \in I_s^c} \rho_{is}^c \sum_{j \in J_i} B_{i,j,t}$$
$$+ R_{st} - D_{st}, \quad \forall s \in S,\ t \in T, \tag{4}$$

$$0 \leq S_{st} \leq C_s, \quad \forall s \in S,\ t \in T, \tag{5}$$

where Vminij and Vmaxij are the minimum and maximum capacity of unit (j) for task (i), respectively; Ips and Ics the set of tasks that produce and consume state (s), respectively;ρ p isand ρc is the fractions of state (s) produced and consumed by task (i), respectively; Ji the set of units suitable for task (i); αis the processing time for state (s) by task (i); and Cs is the storage capacity limit for state (s). Rst is the amount of state (s) received from external sources at time interval (T). Variable Dst represents the amount

of state (s) delivered at time interval (T). Constraints (3) enforce that if task (i) starts in unit (j) at time interval (T) (i.e., Wijt = 1), the batch-size is bounded by the minimum and maximum capacities of the involved unit (Vminij≤ Bijt ≤ Vmaxij). When the task does nottake place (i.e., Wijt = 0), the corresponding batch-size is zero (i.e., Bijt = 0). The mass balance is expressed by Constraint(4), which states that the amount of state (s) during time interval (T) is equal to that during the previous time interval (T − 1) plus the amount produced by tasks that finish at the end of the previous time interval (T − 1) minus the amount consumed by tasks that start at the beginning of the current time interval (T), further adjusted by the amount received from or delivered to external systems at this time interval (T). The restriction on storage of a material state is then represented by Constraint (5).

4. 3 MODEL & RESULTS

Four examples were implemented from literature, the first three based on discrete time representation and the last based on continuous time representation.

CASE STUDY 1:

The example was taken from Shah et al (1993). The problem involved three feeds, four intermediates and two products. The STN for the problem is shown below:

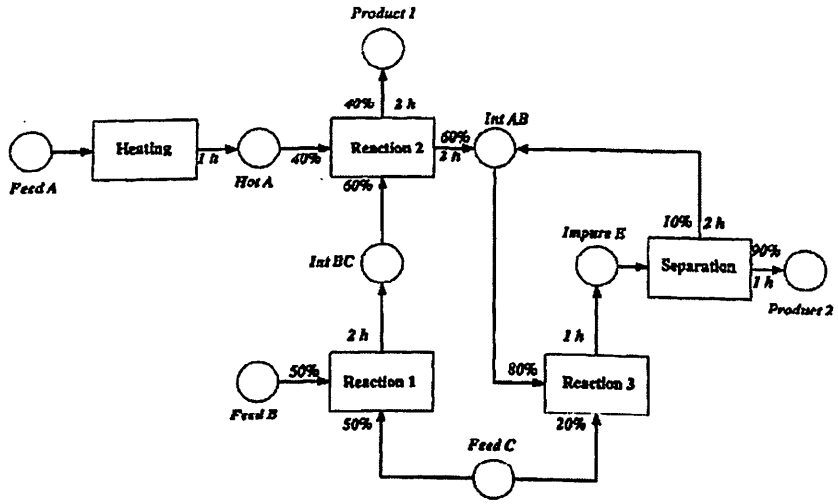

Fig4.2. STN for Case Study 1 (Shah et al, 1993)

The data and the recipe for the problem is given below:

Table 1.Data for Case Study 1

Time horizon: 9 h			
Process Units	**Size**	**Suitability**	**Processing Time, h**
Heater	100	Heating	1
Reactor 1	80	Reaction 1,2,3	2,2,1
Reactor 2	50	Reaction 1,2,3	2,2,1
Still	200	Separation	1 h for product 2, 2h for impure E
Process States	**Storage Capacity,kg**	**Initial Amount**	**Price**
Feeds	Unlimited	Unlimited	0
Hot A	100	0	0
Intermediate AB	200	0	0

Intermediate BC	150	0	0
Impure E	200	0	0
Products 1,2	Unlimited	0	1

The objective function was maximization of profit which was obtained using CPLEX solver in GAMS as 241.5 units which was consistent with the result in the paper.

```
rmatched
Sets

i tasks/ h, r1, r2, r3, s/
j units/heater, reactor1, reactor2, still/
t time points /t0*t5/
s states/A, hotA, B, C, BC, AB, E, p1, p2/
raw(s) raw mat / A, B, C /;

alias(i,ii)
alias(t,tt);

Parameters
C(s) max storage capacity
/A    inf
 B    inf
 C    inf
 hotA 100
 AB   200
 BC   150
 E    200
 p1   inf
 p2   inf/;

Parameter ct(i) completion time for task i
/h       1
 r1      2
 r2      2
 r3      1
 s       2/;

scalar H /9/;
scalar GCF  / 1 /;
scalar int inverval;
int = H / (card(t) * GCF);
```

```
display int;
parameter p(i) no of inverval over which the task i occurs ;
p(i) = CT(i) / int;
display p;


Table
suit(i,j)  suitability for task i in unit j
         heater     reactor1   reactor2     still
h        1
r1                  1          1
r2                  1          1
r3                  1          1
s                                           1;


Table
Vmax(i,j) max storage available for task i in unit j
         heater     reactor1     reactor2     still
h        100
r1                  80           50
r2                  80           50
r3                  80           50
s                                           200;

Table
Vmin(i,j) min storage available for task i in unit j
         heater     reactor1     reactor2     still
h        0          0            0            0
r1       0          0            0            0
r2       0          0            0            0
r3       0          0            0            0
s        0          0            0            0;
```

```
Table
rho(i,s) proportion of input of task i from state s
         A   hotA   B    C    BC    AB    E   p1   p2
h        1
r1                  0.5  0.5
r2           0.4               0.6
r3                       0.2        0.8
s                                        1          ;

Table
rhobar(i,s) proportion of output of task i from state s
         A   hotA   B    C    BC    AB    E   p1   p2
h            1
r1                              1
r2                                    0.6      0.4
r3                                         1
s                                     0.1           0.9;


Table pt(i,s) processing time for output of task i to state s
         A   hotA   B    C    BC    AB    E   p1   p2
h            1
r1                              2
r2                                    2        2
r3                                         1
s                                     2             1;
```

```
Parameter Cost(s) cost data
/
A     3
hotA  5
B     2
C     4
BC    3
AB    2
E     3
p1    1
p2    1/;

variable f objective function;
binary Variables
W(i,j,t) if unit j performs task i in beginning of time interval t;

positive Variable
B(i,j,t) amount of material which undergoes task i in unit j at the beginning of time t

State(s,t) amount of material which is stored in state s at the beginning of time t
State0(s) amount of material which is stored in state s at the beginning of time t
;


Equations

alloc(j,t) allocation constraints
alloc2(i,j,t)
batchconstraintmin(i,j,t) batch size constraint
batchconstraintmax(i,j,t) batch size constraint
materialsizeconstraint(s,t) amt of material stored in state s must not exceed Cs
matbalance(s,t) material balance constraint
matbalance0(s,t) material balance constraint
costfunction objective function;
```

alloc(j,t).. sum(i$(suit(i,j)), W(i,j,t)) =l= 1;

alloc2(i,j,t)$(suit(i,j)).. sum(ii$(suit(ii,j)), sum(tt$(ord(tt)>=ord(t) and ord(tt)<= (ord(t) + p(i)-1)), W(ii,j,tt))) - 1 =l= H * (1 - W(i,j,t));

batchconstraintmin(i,j,t)$(suit(i,j)).. W(i,j,t)* Vmin(i,j) - B(i,j,t) =l= 0;

batchconstraintmax(i,j,t)$(suit(i,j)).. B(i,j,t) -W(i,j,t)* Vmax(i,j) =l= 0;

materialsizeconstraint(s,t).. State(s,t) - C(s) =l= 0;

matbalance(s,t)$(ord(t)>1)..State(s,t)=e=State(s,t-1)+ sum(i,(rhobar(i,s)* (sum(j$(suit(i,j)),(B(i,j,t-p(i)))))))-sum(i,(rho(i,s)*(sum(j$(suit(i,j)),B(i,j,t))))) ;

matbalance0(s,t)$(ord(t)=1)..State(s,t)=e=State0(s)+

```
sum(i,(rhobar(i,s)*(sum(j$(suit(i,j)),(B(i,j,t-p(i)))))))-sum(i,(rho(i,s)*
(sum(j$(suit(i,j)),B(i,j,t))))) ;

costfunction..f=e=sum(s,sum(t$(ord(t)=card(t)),(Cost(s)*State(s,t))))-
sum(s,Cost(s)*State0(s));

*B.lo(i,j,t)=0;
*W.lo(i,j,t)=0;
state0.fx(s)$(not raw(s)) = 0;

Model discrete /all/;

option optcr=0;
option limrow = 1000;
option limcol = 1000;

solve discrete USING MIP maximizing f;
display W.l, B.l, state.l, state0.l, f.l;
```

CASE STUDY 2:

The second case study was also taken from Shah et al (1993) and was the same as the first case study except for the following modifications:

- Time horizon =10 h
- The storage capacities for hot A, intermediate AB and impure E is assumed to be 1000 kg while no storage capacity is available for intermediate BC.
- The price of the two products is set at 10 units.

The objective function was maximization of profit which was obtained using CPLEX solver in GAMS as 2296.25 units which was consistent with the result in the paper.

```
Sets

i tasks/ h, r1, r2, r3, s/
j units heater, reactor1, reactor2, still/
t time points /t1*t11/
s states/A, hotA, B, C, BC, AB, E, p1, p2/
raw(s) raw mat / A, B, C /;

alias(i,ii)
alias(t,tt);

Parameters
C(s) max storage capacity
/A     inf
 B     inf
 C     inf
 hotA  1000
 AB    1000
 BC    0
 E     1000
 p1    inf
 p2    inf/;

Parameter ct(i) completion time for task i
/h      1
 r1     2
 r2     2
 r3     1
 s      2/;

scalar H /10/;
scalar GCF  / 1 /;
scalar int inverval;
int = H / (card(t) * GCF);
display int;
```

```
parameter p(i) no of inverval over which the task i occurs ;
p(i) = CT(i) / int;
display p;


Table
suit(i,j)  suitability for task i in unit j
         heater     reactor1   reactor2     still
h        1
r1                  1          1
r2                  1          1
r3                  1          1
s                                           1;


Table
Vmax(i,j) max storage available for task i in unit j
         heater     reactor1   reactor2     still
h        100
r1                  80         50
r2                  80         50
r3                  80         50
s                                           200;

Table
Vmin(i,j) min storage available for task i in unit j
         heater     reactor1   reactor2     still
h        0          0          0            0
r1       0          0          0            0
r2       0          0          0            0
r3       0          0          0            0
s        0          0          0            0;
```

```
Table
rhobar(i,s) proportion of output of task i from state s
         A   notA   B    C    BC    AB    E   p1   p2
h            1
r1                          1
r2                                0.6       0.4
r3                                     1
s                                 0.1            0.9;


Table pt(i,s) processing time for output of task i to state s
         A   notA   B    C    BC    AB    E   p1   p2
h            1
r1                          2
r2                                2         2
r3                                     1
s                                 2              1;

Parameter Cost(s) cost data
/
A    0
notA 0
B    0
C    0
BC   0
AB   0
E    0
p1   10
p2   10/;

variable f objective function;
binary Variables
W(i,j,t) if unit j performs task i in beginning of time interval t;
```

```
positive Variable
B(i,j,t) amount of material which undergoes task i in unit j at the beginning of time t

State(s,t) amount of material which is stored in state s at the beginning of time t
State0(s) amount of material which is stored in state s at the beginning of time t
;

.

Equations

alloc(j,t) allocation constraints
alloc2(i,j,t)
batchconstraintmin(i,j,t) batch size constraint
batchconstraintmax(i,j,t) batch size constraint
materialsizeconstraint(s,t) amt of material stored in state s must not exceed Cs
matbalance(s,t) material balance constraint
matbalance0(s,t) material balance constraint
costfunction objective function;


alloc(j,t).. sum(i$(suit(i,j)), W(i,j,t)) =l= 1;
```

alloc2(i,j,t)S(suit(i,j)).. sum(iiS(suit(ii,j)), sum(ttS(ord(tt)>=ord(t) and ord(tt)<= (ord(t) - p(i)-1)), W(ii,j,tt))) - 1 =l= H * (1 - W(i,j,t));

batchconstraintmin(i,j,t)S(suit(i,j)).. W(i,j,t)* Vmin(i,j) - B(i,j,t) =l= 0;

batchconstraintmax(i,j,t)S(suit(i,j)).. B(i,j,t) -W(i,j,t)* Vmax(i,j) =l= 0;

materialsizeconstraint(s,t).. State(s,t) - C(s) =l= 0;

matbalance(s,t)S(ord(t)>1)..State(s,t)=e=State(s,t-1)+ sum(i,(rhobar(i,s)* (sum(jS(suit(i,j)),(B(i,j,t-p(i)))))))-sum(i,(rho(i,s)*(sum(jS(suit(i,j)),B(i,j,t))))) ;

matbalance0(s,t)S(ord(t)=1)..State(s,t)=e=State0(s) + sum(i,(rhobar(i,s)* (sum(jS(suit(i,j)),(B(i,j,t-p(i)))))))-sum(i,(rho(i,s)*(sum(jS(suit(i,j)),B(i,j,t))))) ;

```
costfunction.. f =e= sum(s,sum(t$(ord(t)=card(t)),(Cost(s)*State(s,t))))-sum(s, Cost(s)*State0(s));

*B.lo(i,j,t)=0;
*W.lo(i,j,t)=0;
state0.fx(s)$(not raw(s)) = 0;

Model discrete /all/;

option optcr=0;
option limrow = 1000;
option limcol = 1000;


solve discrete USING MIP maximizing f;
display W.l, B.l, state.l, state0.l, f.l;
```

CASE STUDY 3:

This case study was taken from the Appendix of Shah et al (1993) and involved three feeds, three intermediates and three products. The STN for the problem is as shown:

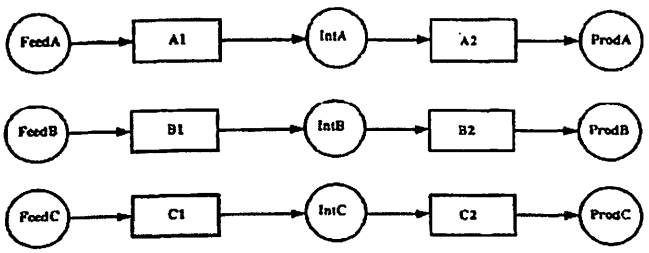

Fig.4.3. STN for Case Study 3 (Shah et al,1993)

The data and the recipe for the problem is shown below:

Table 2: Data for Case Study 3

Time Horizon:10 h			
Process Units	**Size**	**Suitability**	**Task Processing Times (h)**
Unit 1	2029	A1	2
Unit 2	1691	A2,C1	1,1
Unit 3	720	B1,C2	1,2
Unit 4	929	B2	1
Process States	**Storage Capacity**	**Initial Amount**	**Price**
Feeds	Unlimited	Unlimited	0
Intermediate B, Intermediate C	10,000	0	0
Intermediate A	0	0	0
Products A,B,C	Unlimited	0	1,1,2.5

The objective function was maximization of profit which was obtained using CPLEX solver in GAMS as 16,756 units which was consistent with the result in the paper.

```
Sets

i tasks/ a1, a2, b1, b2, c1, c2/
j units/u1, u2, u3, u4/
t time points /t1*t11/
s states/A, B, C, intA, intB, intC, pA, pB, pC/
raw(s) raw mat / A, B, C /;

alias(i,ii)
alias(t,tt);

Parameters
C(s) max storage capacity
/A     inf
 B     inf
 C     inf
 intA 10000
 intB 100
 intC 100
 pA    inf
 pB    inf
 pC    inf
/;
Parameter ct(i) completion time for task i
/a1      2
a2       1
b1       1
b2       1
c1       1
c2       2
/;
scalar H /10/;
scalar GCF  / 1 /;
scalar int inverval;
int = H / (card(t) * GCF);
display int;
```

```
Table
suit(i,j)  suitability for task i in unit j
         u1       u2       u3       u4
a1       1
a2                1
b1                         1
b2                                  1
c1                1
c2                         1
;

Table
Vmax(i,j) max storage available for task i in unit j
         u1       u2       u3       u4
a1       2029
a2                1691
b1                         720
b2                                  929
c1                1691
c2                         720
;

Table
Vmin(i,j) max storage available for task i in unit j
         u1       u2       u3       u4
a1
a2
b1
b2
c1
c2
;
```

```
Table
rho(i,s) proportion of input of task i from state s
         A       B       C       intA    intB    intC    pA      pB      pC
a1       1
a2                               1
b1               1
b2                                       1
c1                       1
c2                                               1
;

Table
rhobar(i,s) proportion of output of task i from state s
         A       B       C       intA    intB    intC    pA      pB      pC
a1                               1
a2                                                       1
b1                                       1
b2                                                               1
c1                                               1
c2                                                                       1;



Table pt(i,s) processing time for output of task i to state s
         A       B       C       intA    intB    intC    pA      pB      pC
a1                               2
a2                                                       1
b1                                       1
b2                                                               1
c1                                               1
c2                                                                       2;
```

```
Parameter Cost(s) cost data

A       1
B       1
C       1
intA    0
intB    0
intC    0
pA      1
pB      1
pC      2.5
/;

variable f objective function;
binary Variables
W(i,j,t) if unit j performs task i in beginning of time interval t;

positive Variable
B(i,j,t) amount of material which undergoes task i in unit j at the beginning of time t

State(s,t) amount of material which is stored in state s at the beginning of time t
State0(s) amount of material which is stored in state s at the beginning of time t
;


Equations

alloc(j,t) allocation constraints
alloc2(i,j,t)
*back(j,t)
batchconstraintmin(i,j,t) batch size constraint
batchconstraintmax(i,j,t) batch size constraint
materialsizeconstraint(s,t) amt of material stored in state s must not exceed Cs
matbalance(s,t) material balance constraint
matbalance0(s,t) material balance constraint
costfunction objective function;
```

```
alloc(j,t).. sum(iS(suit(i,j)), W(i,j,t)) =l= 1;

alloc2(i,j,t)S(suit(i,j)).. sum(iiS(suit(ii,j)), sum(ttS(ord(tt)>=ord(t) and ord(tt)<= (ord(t) +
p(i)-1)), W(ii,j,tt))) - 1 =l= H * (1 - W(i,j,t));

*back(j,t).. sum(iS(suit(i,j)), sum(ttS(ord(tt)>=ord(t) and ord(tt)<= ord(t)-p(i)+1),
W(i,j,tt) ) ) =l= 1;

batchconstraintmin(i,j,t)S(suit(i,j)).. W(i,j,t)* Vmin(i,j) - B(i,j,t) =l= 0;
batchconstraintmax(i,j,t)$(suit(i,j)).. B(i,j,t) -W(i,j,t)* Vmax(i,j) =l= 0;

materialsizeconstraint(s,t).. State(s,t) - C(s) =l= 0;

matbalance(s,t)S(ord(t)>1)..State(s,t) =e= State(s,t-1) + sum(i,(rhobar(i,s)*
(sum(jS(suit(i,j)),(B(i,j,t-p(i)))))))-sum(i,(rho(i,s)*(sum(jS(suit(i,j)),B(i,j,t))))) ;

matbalance0(s,t)$(ord(t)=1)..State(s,t)=e=State0(s)+ sum(i,(rhobar(i,s)*
(sum(jS(suit(i,j)),(B(i,j,t-p(i)))))))-sum(i,(rho(i,s)*(sum(jS(suit(i,j)),B(i,j,t))))) ;

costfunction.. f =e= sum(s,sum(t$(ord(t)=card(t)),(Cost(s)*State(s,t))))-sum(s, Cost(s)*State0(s));


state0.fx(s)$(not raw(s)) = 0;

Model discrete /all/;

option optcr=0;
option limrow = 1000;
option limcol = 1000;


solve discrete USING MIP maximizing f;
display W.l, B.l, state.l, state0.l, f.l;
```

CASE STUDY 4:

This case study was taken from Ierapetritou & Floudas (1998). It consists of four states and three units and the STN is as shown below. The model was implemented using continuous time formulation using the constraints mentioned in the same paper.

Mathematical Formulation

The proposed formulation focuses around the following key ideas:

(a) Continuous time representation: The proposed formulation is based on a continuous time representation that avoids the prepostulation of unnecessary time intervals. It only requires the initial consideration of a necessary number of event points corresponding to either the initiation of a task or the beginning of unit utilization. The location of these points is unknown. (b) Decoupling of task events from unit events : The basic idea of the proposed formulation is that it decouples the tusk events (i) from the unit events (j). This is ac hievedby the consideration of different variables to represent the task events (i.e., the beginning of the task), denoted as ulv(i, n), and the unit events (i.e., the beginning of unit utilization), denoted as yv(j, n). If task event (i) starts at event point (n) then wv(i,n)=l, otherwise it is zero. If unit event (i) takes place at event point (n), then yv(i,n)= 1, otherwise it is zero.(c) Variable Processing times: Processing times are considered to vary with respect to the amount of the material being processed by the specific task.The mathematical model for the short-term scheduling of batch plants involves the following constraints:

Allocation Constraints

$$\sum_{i \in I_j} wv(i,n) = yv(j,n), \quad \forall j \in J, \quad n \in N \qquad (1)$$

These constraints express that at each unit (j) and at a event point(n) only one of the tasks that can be performed in this unit (i.e., i E 1j) should take place. If unit(i) is utilized at event point (n), that is, yv(i,n) equal 1, then one of the wv(i,n) variables should be activated. If unit (j) is not utilized at point (n), then all wv(i,n) variables take zero values,

that is no assignments of tasks are made

Capacity Constraints

$$V_{ij}^{min} wv(i,n) \leq B(i,j,n) \leq V_{ij}^{max} wv(i,n), \quad \forall i \in I,\ j \in J_i,\ n \in N \qquad (2)$$

where B(i, j, n) correspond to the amount of material undertaking task (i) in unit (j) at event point (n). These constraints express the requirement for minimum amount, V.Fin, of material in order for a unit (j) to start operating ta!k (i), and the maximum capacity of a unit (i), Viy'z, when performing task (i). If wv(i,n) equals one, then constraints (2) correspond to lower and upper bounds on the capacities B(ij,n). If wv(i,n) equals zero, then all B(ij,n) variables become zero.

Storage Constraints

$$ST(s,n) \leq ST(s)^{max},\ \forall s \in S,\ n \in N \qquad (3)$$

where ST(s, n) corresponds to the amount of material (s) at event point (n). These constraints represent the maximum available storage capacity for each state (s), at each event point (n).

Material Balances

$$ST(s,n) = ST(s,n-1) - d(s,n) + \sum_{i \in I_s} \rho_{si}^{p} \sum_{j \in J_i} B(i,j,n-1) + \sum_{i \in I_s} \rho_{si}^{c} \sum_{j \in J_i} B(i,j,n), \quad \forall s \in S,\ n \in N \qquad (4)$$

where $\rho_{si}^{c} \leq 0, \rho_{si}^{p} \geq 0$ Represent the proportion of state (s) consumed or produced from task (i), respectively. According to these constraints the amount of material of state (s) at event point (n) is equal to that at event point (n-l) adjusted by any amounts

produced or consumed between the event points (n-l) and (n) and the amount required by the market at event point(n) within the time horizon.

Duration Constraints

$$T^f(i,j,n) = T^s(i,j,n) + \alpha_{ij} wv(i,n) + \beta_{ij} B(i,j,n) \quad \forall i \in I,\ j \in J_i,\ n \in N \qquad (6)$$

Sequence Constraints: Same task in the same unit

$$T^s(i,j,n+1) \geq T^f(i,j,n) - H(2 - wv(i,n) - yv(j,n)) \quad \forall i \in I,\ j \in J_i,\ n \in N, n \neq N \qquad (7)$$

The sequence constraints (7) state that task (i) starting at event point (n+l) should start after the end of the same task performing at the same unit (i) which has already started at event point (n). If task (i) takes place in unit (j) at event point (n) (i.e., wv(i,n)=yv(i,n)=l), then we have the second term of (7) become zero. If either wv(i,n) or yv(j,n) or both are equal to zero, then constraint (7) is relaxed. In a similar manner sequence constraints are introduced for different tasks that can take place in the same or different units (Ierapetritou and Floudas, 1997a.b).

Objective: Maximization of profit

$$\sum_s \sum_n \text{price}(s) d(s,n) \qquad (8)$$

The objective shown in (8) is the maximization of production in terms of profit due to product sales .

Fig.4.4. STN for Case Study 4

The data and recipe for the problem is as shown below:

Table 3: Data for Case Study 4

Time Horizon:12 h			
Units	**Capacity**	**Suitability**	**Mean Processing Time (h)**
Unit 1	100	Task 1	4.5
Unit 2	75	Task 2	3
Unit 3	50	Task 3	1.5
States	**Storage Capacity**	**Initial Amount**	**Price**
State 1	Unlimited	Unlimited	0
State 2	100	0	0
State 3	100	0	0
State 4	Unlimited	0	1

The objective function was maximization of profit which was obtained using CPLEX solver in GAMS as 71.518 units which was consistent with the result in the paper.

```
set
i tasks /t1*t3/
n time points /n0*n4/
j units /u1*u3/
s states /s1*s4/
raw(s) /s1/
;
alias(i,ii);

alias(j,jj);

alias(n,nn);
parameter Vmin(i,j);
Vmin(i,j)=0;

table suit(i,j)
    u1   u2   u3
t1  1
t2       1
t3            1
;

table Vmax(i,j)
    u1   u2   u3
t1  100
t2       75
t3            50
;

parameter STmax(s)
/
s1   inf
s2   100
s3   100
s4   inf
/;
```

```
table rhoP(s,i) proportion of state produced from task i
         t1       t2       t3
s1
s2       1
s3                1
s4                         1
;

table rhoC(s,i) proportion of state produced from task i
         t1       t2       t3
s1       1
s2                1
s3                         1
s4
;

table alpha(i,j) constant term of processing task i at unit j
   u1     u2      u3
t1 3
t2       2
t3               1
;

table beta(i,j) constant term of processing task i at unit j
   u1         u2          u3
t1 0.03
t2            0.0266
t3                        0.02
;

scalar H /12/;
```

```
parameter price(s) price of state s
/
s1        3
s2        3
s3        3
s4        1
/;

binary variable
wv(i,n) binary variable that assign the beginning of task i at event point n
yv(j,n) binary variable that assign the utilisation of unit j at event point n
;
positive variable
B(i,j,n) amount of material undergoing task i to unit j at event point n
ST(s,n) amount of state s at event point n
Ts(i,j,n) time that task i starts in unit j at event point n
Tf(i,j,n) time that task i finishes in unit j while it starts at event point n
ST0(s) initial state of s
d(s,n) demand of state s at time point n
;

variable f;

equations
alloc(j,n)
capacitymin(i,j,n)
capacitymax(i,j,n)
storage(s,n)
matbalance(s,n)
matbalance0(s,n)
duration(i,j,n)

seq_ss1(i,j,n) constraints for same task in same unit
seq_ss2(i,j,n)
seq_ss3(i,j,n)
```

*seq_ds(ii,i,j,n) different task in same unit

seq_dd(ii,jj,i,j,n) different task in different unit

seq_comp(i,j,n) completion of previous task

horizon_cons1(i,j,n)

horizon_cons2(i,j,n)

obj

;

```
alloc(j,n).. sum(iS(suit(i,j)), wv(i,n) ) =e= yv(j,n);
capacitymin(i,j,n)S(suit(i,j)).. Vmin(i,j)*wv(i,n) =l= B(i,j,n);
capacitymax(i,j,n)S(suit(i,j)).. B(i,j,n) =l= Vmax(i,j)*wv(i,n);
storage(s,n).. ST(s,n) =l= STmax(s);
matbalance(s,n)S(ord(n)>1)..ST(s,n)=e=ST(s,n-1)+ sum(i,(rhoP(s,i)*
(sum(jS(suit(i,j)),(B(i,j,n-1))))))-sum(i,(rhoC(s,i)*(sum(jS(suit(i,j)),B(i,j,n)))))-d(s,n) ;
matbalance0(s,n)S(ord(n)=1)..ST(s,n)=e=ST0(s)+ sum(i,(rhoP(s,i)*
(sum(jS(suit(i,j)),(B(i,j,n-1))))))-sum(i,(rhoC(s,i)*(sum(jS(suit(i,j)),B(i,j,n))))) ;
duration(i,j,n).. Tf(i,j,n) =e= Ts(i,j,n)+alpha(i,j)*wv(i,n)+beta(i,j)*B(i,j,n);

seq_ss1(i,j,n)S( ord(n)<card(n) S(suit(i,j)) ).. Ts(i,j,n+1) =g= Tf(i,j,n) -H*(2-wv(i,n)-
yv(j,n));
seq_ss2(i,j,n)S(ord(n)<card(n)S(suit(i,j))).. Ts(i,j,n+1) =g= Ts(i,j,n);
seq_ss3(i,j,n)S(ord(n)<card(n)S(suit(i,j))).. Tf(i,j,n+1) =g= Tf(i,j,n);
```

```
seq_dd(ii,jj,i,j,n)$( (ord(n)<card(n) and ord(i) ne ord(ii) )$(suit(i,j) )).. Ts(i,j,n+1) =g= Tf(ii,jj,n)-H*(2-wv(ii,n)-yv(jj,n));

seq_comp(i,j,n)$(ord(n)<card(n)$(suit(i,j))).. Ts(i,j,n+1) =g= sum(nn$(ord(nn)<=ord(n)) , sum(ii ,(Tf(ii,j,nn) -Ts(ii,j,nn)) ) );

horizon_cons1(i,j,n)$(suit(i,j)).. Tf(i,j,n) =l= H;

horizon_cons2(i,j,n)$(suit(i,j)).. Ts(i,j,n) =l= H;

obj..f =e= sum(s, sum(n, price(s)*d(s,n) ) );
i.fx(s,n)$(ord(s)=4 and ord(n)=1 )=0;

ST0.fx(s)$(not raw(s))=0;

option optcr=0;

model continuous /all/;
solve continuous using MIP maximising f;
display f.l, d.l, ST.l, wv.l, yv.l, Tf.l, Ts.l,B.l;
```

CHAPTER 5
SUMMARY / CONCLUSIONS

Discrete & continuous time formulations were implemented from benchmark examples from literature. The models were based on the constraints given in Shah et al (1993) for discrete time models and on Ierapetritou & Floudas (1998) for continuous time models. The results obtained were consistent with that found in literature. Model statistics for the case study are summarized below:

Table 4: Summary of Case Study Results

CASE STUDY	**1**	**2**	**3**	**4**
Discrete variables	48	88	66	30
Single variables	160	285	241	177
Single Equations	277	507	441	281
Objective value	241.5000	2296.2500	16756.0000	71.5182
Solver status	Normal Completion	Normal Completion	Normal Completion	Normal Completion

5.1 SCOPE OF FUTURE WORK

Future work would involve:

- Implementation of unit-specific continuous time formulations from research papers showing an improvement on the Ierapetritou & Floudas (1998) model
- Implementation of reactive scheduling formulations from research papers
- Development of a reactive scheduling model as an improvement on previous existing models

Chapter 6
Brief Introduction to GAMS

GAMS is a high level modeling system with following components :-

Components

- Language compiler
- Integrated high performance solvers

Models supported

- Linear Programs
- Non-Linear Programs
- Mixed Integer Optimization

Structure of GAMS model comprises of the following:-

INPUTS

Set

- Declaration
- Assignment - indices

Data (Parameters, Tables, Scalars)

- Declaration
- Assignment - values

Variables

- Declaration
- Type Assignment
- Assignments of Bounds (optional)

Equations

- Declaration
- Definition

Model and Solve Statements

- Display Statements (optional)

OUTPUT

Echo Print

- Errors (if present)
- Reference Maps
- Equation Listings
- Status Reports
- Results

Sets

- Sets are the basic building blocks of a GAMS model, corresponding exactly to the indices in the algebraic representations of models.

Different ways of declaration sets :-

```
i supply points /1, 2/
j demand points /1, 2, 3/;
set j demand points /1*3/;
```

- Alias(i,k)
- Multidimensional sets
- Dynamic Sets

Data

Different formats for data entry

- Lists
- Tables
- Direct Assignments
- Assignments from files

Data entry by lists

- parameter d(j) demand at point j

/1 100
2 200
3 300/;

- List must be enclosed within slashes, entries separated by commas or separate lines
- Domain Checking
- Default value is zero
- Scalar is considered as a parameter with no domain
- Parameter's with multi-dimensional domains can also be entered by lists

- Data entry by tables

table dist(i,j) distance from i to j

	1	2	3
1	10.2	5.3	50
2	30	21.2	4.5;

- Declares parameter dist and specifies its domain as set of ordered pairs of Cartesian product of i, j
- Blanks are interpreted as zero
- Domain checking
- Entering tables with more than one dimension

Data entry by direct assignment

- parameter c(i,j) cost of transportation from i to j;

c(i,j) = M * dist_1(i,j)*10;

- Semicolon between two statements
- Parameters used in computation have to be previously declared
- Domain checking
- Specific elements in the domain can be assigned by

c('1','3') = 100;

Variables

Decision variables must be declared

Variables

x(i,j) shipment quantities in cases

z total transportation costs in thousands of dollars ;

- Default type is free
- Different permissible types for variables

Variable Type	Allowable Range of Variables
free (default)	- ∞ to +∞
Positive	0 to +∞
Negative	-∞ to 0
Binary	0 or 1
Integer	0,1,2,..etc

Summation and Product in GAMS

- Not possible to have standard mathematical notation for summation and product
- Summation
  - sum(index of summation, summand)
  - sum(j, x(i,j)) is equivalent to
  - sum((i,j), c(i,j)*x(i,j)) is equivalent to
- Product
  - prod(j, x(i, j)) is equivalent to

- Product and Summation can be used to make assignments and in implementing equations
- Conditional summation and product

Equations

- Must be declared and defined in separate statements
- Declaration :
    - equations obj, const1, const2;
- Definition :
    - The name of the equation being defined
    - The domain
    - Domain restriction condition (optional)
    - The symbol '..'
    - Left-hand-side expression
    - Relational operator: =l=, =e=, or =g=
    - Right-hand-side expression
- The transportation example contains three of these statements.
    - cost .. z =e= sum((i,j), c(i,j)*x(i,j)) ;
    - supply(i) .. sum(j, x(i,j)) =l= a(i) ;
    - demand(j) .. sum(i, x(i,j)) =g= b(j) ;

GAMS OUTPUT

- Echo Print
- Errors (if present)
    - Example **** $160
    - Summary of the error messages is given at the end of the filename.lst file
- Reference Maps
- Equation Listings
- Status Reports
- Result

CHAPTER 7
REFERENCES

1. Floudas, C.A & Xiaoxia Lin *(2004) Continuous-time versus discrete-time approaches forscheduling of chemical processes: a review* , Computers and Chemical Engineering 28 (2004) 2109–2129
2. Christos T. Maravelias, Ignacio E. Grossmann (2002) *A general Continuous –time state task network formulation for short term scheduling of multipurpose batch plants.* Pdf article
3. Ierapetritou, M.G. & Floudas, C.A. (1998) *Effective Continuous Time Formulation for short term scheduling: I. Multipurpose Batch Process* ,Industrial & Engineering Chemistry Research., 37 (11), 4341–4359
4. Janak, S. L. & Floudas, C.A. (2006) *Production Scheduling of a Large-scale Industrial Batch Plant II. Reactive Scheduling,* Industrial & Engineering Chemistry Research., 45, 8253-8269
5. Kanakamedala, K. B., Reklaitis, G.V.& Venkatasubramanian, V. (1994) *Reactive Schedule Modification in Multipurpose Batch Chemical Plants,* Industrial & Engineering Chemistry Research ,33, 77-90
6. Mendez, C.A. & Cerda, J. (2003) *Dynamic scheduling in multiproduct batch plants* ,Computers and Chemical Engineering, 27, 1247- 1259
7. Mendez, C.A., & Cerda, J. (2004) *An MILP framework for batch reactive scheduling with limited discrete resources,* Computers and Chemical Engineering., 28, 1059-1068
8. Sanmarti, E., Huercio,A., Espuna,A. & Puigjaner, L. (1996) *A combined scheduling/reactive scheduling strategy to minimize the effect of process operations uncertainty in batch plants,* Computers & Chemical Engineering, 20, 1263-1268
9. Shah, N., Pantelides, C.C. & Sargent, R.W.H., (1993) *A General Algorithm for Short Term Scheduling of Batch Operations-II Computational Issues* , Computers & Chemical Engineering. , 17(2), 229-244

10. Jia, Z. and Ierapetritou. M. (2004). Efficient short-term scheduling of refinery operations based on a continuous time formulation. Computers and Chemical Engineering, 28, 1001–1019.
11. Jia, Z. and Ierapetritou, M.(2003). Refinery Short-Term Scheduling Using Continuous Time Formulation: Crude-Oil Operations. *Industrial and Engineering Chemistry Research*, *42*, 3085-3097.
12. Kondili, E., Pantelides, C.C., and Sargent, W. H. (1993). A General Algorithm for Short-term Scheduling of Batch Operatioms-I. Milp Formulations. Computers and Chemical Engineering, 17, 211-227.
13. Mendez, C.A., Cerd, J., Grossmann, I.E., Iiro, H. I., and Fahl, C. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers and Chemical Engineering, 30, 913–946.
14. Jia, Z. and Ierapetritou, M (2003) . Mixed integer linear programming model for gasoline blending and distribution scheduling