Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers and Chemical Engineering 28 (2004) 2109–2129

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Review

Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review

Christodoulos A. Floudas*, Xiaoxia Lin

Department of Chemical Engineering, School of Eng/Applied Science, The Engineering Quadrangle, Princeton University, Princeton, NJ 08544-5263, USA

Received 11 September 2002; received in revised form 27 March 2003; accepted 25 March 2004

Available online 13 July 2004

Abstract

An overview of developments in the scheduling of multiproduct/multipurpose batch and continuous processes is presented. Existing approaches are classified based on the time representation and important characteristics of chemical processes that pose challenges to the scheduling problem are discussed. In contrast to the discrete-time approaches, various continuous-time models have been proposed in the literature and their strengths and limitations are examined. Computational studies and applications are presented. The important issues of incorporating scheduling at the design stage and scheduling under uncertainty are also reviewed.

Keywords: Process scheduling; Multiproduct plants; Multipurpose plants; Discrete-time model; Continuous-time model; Mixed-integer linear programming (MILP); Mixed-integer nonlinear programming (MINLP); Design and scheduling; Uncertainty; Reactive scheduling

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 (Pekny & Reklaitis, 1998). These problems are known to belong to the set of NP-complete problems (Garey & Johnson, 1979). As a result, all existing algorithms scale exponentially in the worst case. Therefore, a modest growth in problem size can lead to a significant increase in the computational requirements. This has important implications for the solution of scheduling problems.

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. Bassett, Dave, et al. (1996) presented an overview of existing strategies for implementing integrated applications based on mathematical programming models and examined four classes of integration including scheduling, control, planning and scheduling across single and multiple sites, and design under uncertainty. Applequist, Samikoglu, Pekny, and Reklaitis (1997) discussed the

* Corresponding author. Tel.: +1 609 258 4595; fax: +1 609 258 0211.
E-mail address: floudas@titan.princeton.edu (C.A. Floudas).

0098-1354/$ – see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2004.05.002

Nomenclature

α_{ij}	parameter, fixed processing time of task (i) in unit (j)
α_{is}	parameter, processing time for state (s) by task (i)
β_{ij}	parameter, linear coefficient of the variable term of the processing time of task (i) in unit (j)
ρ_{is}^{p}, ρ_{is}^{c}	parameters, fractions of state (s) produced and consumed by task (i), respectively
$\tau_{ii'j}$	parameter, duration of the cleaning operation required for unit (j) to switch from order (i) to (i')
$\tau_{jff'}$	parameter, duration of the cleaning operation required for unit (j) to switch from product family (f) to (f')
$\tau_{ji'i}$	parameter, duration of the cleaning operation required for unit (j) to switch from task (i') to (i)
τ_k	continuous variable, durations of time slot (k)
B_{ijt}	continuous variable, amount of material which starts undergoing task (i) in unit (j) at time interval (t)
$B(i, j, n)$	continuous variable, amount of material which starts undergoing task (i) in unit (j) at event point (n)
C_{il}	continuous variable, completion time of order (i) at stage (l)
C_{s}	parameter, storage capacity limit for state (s)
d_i	parameter, due date of order (i)
D_{st}	continuous variable, amount of state (s) delivered at time interval (t)
$D(s, n)$	continuous variable, amounts of state (s) delivered at event point (n)
da_{sn}	parameter, amount of the demand for state (s) at event point (n)
dd_{sn}	parameter, due date of the demand for state (s) at event point (n)
f, f'	indices, product families
f	set of product families
H	parameter, time horizon
i, i'	indices, tasks or orders
$i_{jff'}$	index, cleaning task required for unit (j) to switch from product family (f) to (f')
I	set of all tasks or orders
I_j	set of tasks or orders that can be performed in unit (j)
$I_j^{(f)}$, $I_j^{(f')}$	sets of tasks for family (f) and (f') that can be performed in unit (j), respectively
I_s^{p}, I_s^{c}	sets of tasks that produce and consume state (s), respectively
j, j'	index, units
J	set of all units
J_i	set of units suitable for task or order (i)
J_l	set of units for stage (l)
J_{il}	set of units for stage (l) of order (i)
$J_{ii'l}$	set of units for stage (l) of both orders (i) and (i')
k, k'	indices, time slots or events
K	set of events
K_j	set of time slots of unit (j)
l	index, stage
l_i^l	index, the last stage for order (i)
L_i	set of stages included in the processing of order (i)
L_j	set of stages that can be performed in unit (j)
$L_{ii'}$	set of stages included in the processing of both orders (i) and (i')
M	parameter, a sufficiently large positive number
n	index, event point
N	set of event points
n_{last}	index, the last event point
p	index, time point
pt_{ij}	parameter, processing time for order (i) in unit (j)
R_{st}	parameter, amount of state (s) received from external sources at time interval (t)
R_{ij}^{min}, R_{ij}^{max}	parameters, minimum and maximum processing rates of unit (j) when performing task (i), respectively
ro_i	parameter, earliest time at which order (i) can start
ru_j	parameter, earliest available time of unit (j)
s	index, material state
S	set of material states

S_{st}	continuous variable, amount of material state (s) during time interval (t)
SL(s, n)	continuous variable, difference between the amount of the demand for state (s) and the amount of state (s) delivered at event point (n)
su$_{ij}$	parameter, setup time for order (i) in unit (j)
ST(s, n)	continuous variable, amount of state (s) stored at event point (n)
t, t', t_1, t_2	indices, time intervals
t_{ijk}	continuous variable, duration of task (i) which starts at T_k in unit (j)
T	set of time intervals
T_k	continuous variable, timing of event (k)
$T^{\mathrm{s}}(i, j, n)$, $T^{\mathrm{f}}(i, j, n)$	continuous variables, starting and ending times of task (i) in unit (j) at event point (n), respectively
Ts$_{jk}$	continuous variable, the starting time of time slot (k) of unit (j)
Tsi$_{il}$, Tei$_{il}$	continuous variables, the starting and completion times of stage (l) of order (i)
$X_{ii'l}$	binary variable, whether or not order (i) precedes order (i') at stage (l)
$X_{ijkk'}$	binary variable, whether or not task (i) starts at T_k in unit (j) and completes at $T_{k'}$
$y(s, p)$	binary variable, whether or not state (s) is used at time point (p)
Y_{ij}	binary variable, whether or not to allocate order (i) to unit (j)
$Y_{ikk'}$	binary variable, whether or not task (i) starts at time (k) and is active over slot (k') $\geq$ (k)
yv(j, n)	binary variable, whether or not unit (j) starts being utilized at event point (n)
$V_{ij}^{\min}$, $V_{ij}^{\max}$	parameters, minimum and maximum capacity of unit (j) for task (i), respectively
W_{ijkl}	binary variable, whether or not stage (l) of order (i) is assigned to time slot (k) of unit (j)
W_{ijt}	binary variable, whether or not task (i) starts in unit (j) at the beginning of time interval (T)
W_{ijk}	binary variable, whether or not task (i) starts at T_k in unit (j)
wv(i, n)	binary variable, whether or not task (i) starts at event point (n)

formulation and solution of process scheduling and planning problems, as well as issues associated with the development and use of scheduling software. Shah (1998) examined first different techniques for optimizing production schedules at individual sites, with an emphasis on formal mathematical methods, and then focused on progress in the overall planning of production and distribution in multi-site flexible manufacturing systems.

Pekny and Reklaitis (1998) discussed the nature and characteristics of the scheduling/planning problems and pointed out the key implications for the solution methodology for these problems. They reviewed the available scheduling technologies, including randomized search, rule-based methods, constraint guided search, simulation-based strategies, as well as mathematical programming formulation based approaches using conventional and engineered solution algorithms. Pinto and Grossmann (1998) presented an overview of assignment and sequencing models used in process scheduling with mathematical programming techniques. They identified two major categories of scheduling models—one for single-unit assignment and the other for multiple-unit assignment—and discussed the critical issues of time representation and network structure.

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. The main objective of this review paper is to provide an overview of the discrete-time and continuous-time models for chemical process scheduling. The rest of this paper is organized as follows. First, the mathematical models are classified based on the time representation and we discuss the major characteristics and challenges of the process scheduling problems. Then, the discrete-time approach is presented. Subsequently, the various continuous-time models that have been proposed in the literature are presented along with their strengths and limitations. A summary of computational studies and applications follows. Finally, two important issues of integrated design, synthesis and scheduling, and scheduling under uncertainty are presented.

2. Classification of scheduling formulations

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 will be given in Section 3.

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. A detailed examination of the existing continuous-time formulations for various process scheduling problems will be provided in Section 4.

2.2. Characteristics of process scheduling problems

There are a number of common components involved in most process scheduling problems, such as equipment-task assignment, sequencing and timing of activities. Nevertheless, different problems may also vary significantly in the following aspects, which present different requirements or degrees of difficulty for the modeling of these processes.

2.2.1. Processing sequences

Based on the complexity of processing sequences employed to produce products, we classify all the processes in multiproduct/multipurpose plants into two different groups:

- *Sequential processes*: Different products follow the same processing sequence. It is usually possible to define processing stages, which can be single stage or multiple stages. There can be only one unit per stage or parallel units at each stage. For this type of process, batches are used to represent production and it is thus not necessary to consider mass balances explicitly.
- *Network-represented processes*: When production recipes become more complex and/or different products have low recipe similarities, processing networks are used to represent the production sequences. This corresponds to the more general case in which batches can merge and/or split and material balances are required to be taken into account explicitly. Kondili, Pantelides, and Sargent (1993) proposed a general framework of State-Task Network (STN) for the ambiguity-free representation of such processes. The STN representation of a chemical process is a directed graph with two types of distinctive nodes: the *state* nodes denoted by a circle, representing raw materials, intermediate materials or final products, and the *task* nodes denoted by a rectangle box, representing an operation. The fraction of a state consumed or produced by a task, if not equal to one, is given beside the arch linking the corresponding state and task nodes. As an example, Fig. 2 gives

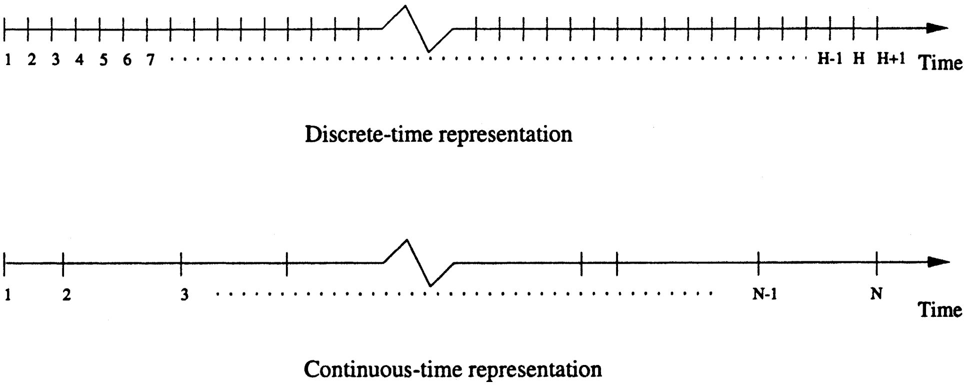

Fig. 1. Discrete and continuous representations of time.

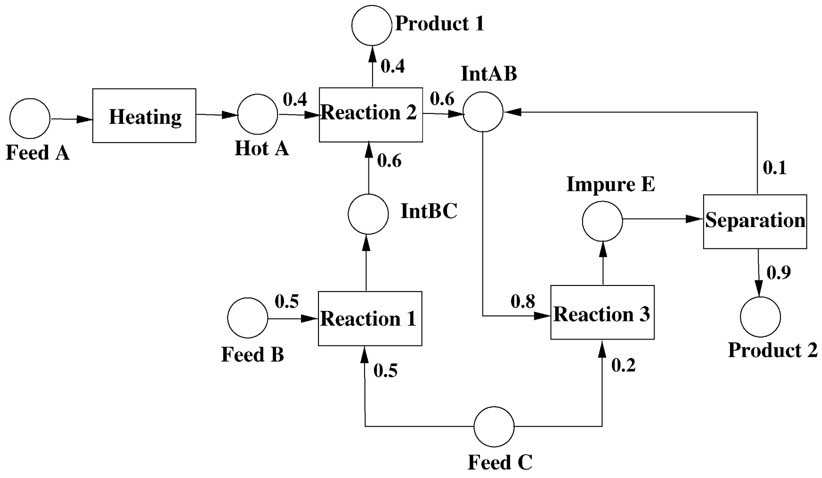

Fig. 2. Example of State-Task Network.

the STN representation of a process involving the merging/splitting of materials/batches and recycle that have been widely studied in the literature. Pantelides (1993) extended the STN to the Resource-Task Network (RTN) framework, which describes processing equipment, storage, material transfer and utilities as resources in a unified way. The RTN representation of the same process as in the STN example is provided in Fig. 3. In addition to the resources of materials, denoted also by circles, the related four pieces of equipment, denoted by ellipses, are also included. Tasks taking place in different units are now treated as different tasks.

2.2.2. Intermediate storage policies

There exist four major categories of treating intermediate storage.

- *Unlimited intermediate storage (UIS)*: In this case, there is no need to model inventory levels.
- *No intermediate storage (NIS)*: There are no storage tanks available for intermediate materials. However, the materials can be held in the processing unit after the task is finished before they are transfered into the next unit.
- *Zero-wait (ZW)*: Relevant intermediate materials are required to be consumed immediately after being produced. Special timing constraints are required to be incorporated.
- *Finite intermediate storage (FIS)*: This correspond to the most general case.

2.2.3. Changeovers

There exist three main types of changeovers.

- *Sequence dependent*: When switched between tasks, a unit may require clean-up or setup for safety or quality rea-

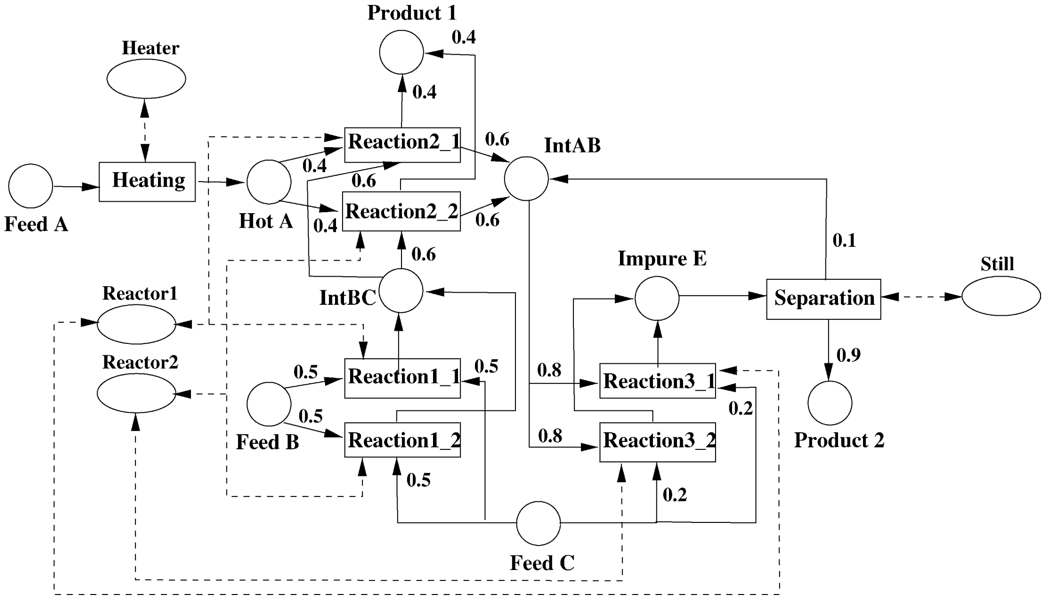

Fig. 3. Example of Resource-Task Network.

sons. The requirement depends on the unit and the tasks involved.

- *Time or frequency dependent*: A changeover may be needed after a certain amount of time or a certain number of batches (tasks).
- *None*: No changeover is needed between two tasks in a unit.

2.2.4. Operation modes of processing tasks

The processing tasks can be classified into batch and continuous tasks.

- *Batch task*: Materials are fed at the start of the task; after a certain period of time, products are produced at the end of the task.
- *Continuous task*: Materials are fed and/or products are produced continuously during the course of the task. The processing rate can either be fixed or within a certain range.

2.2.5. Demand patterns

There exist two main classes of demand patterns.

- *Demands due at the end of horizon*: Demands for products are specified at the end of the horizon under consideration.
- *Demands due at intermediate dates*: Demands for products are specified at designated time instances within the time horizon.

2.2.6. Resource considerations

There are two primary types of resource considerations.

- *Renewable resources*: The operations may require utilities, such as steam, cooling water, electricity, and/or manpower. These are regarded as renewable resources which are completely recovered at the time a task finishes. These resources can never exceed the maximum availability at any time during the production.
- *None*: No restrictions on resources are considered.

2.2.7. Objectives

Typical examples of overall objectives in process scheduling problems include:

- *Minimize makespan*: Given the production requirement, the objective is to find the optimal schedule with the shortest completion time of the whole process.
- *Minimize earliness/tardiness/costs*: Given the production requirement, the optimal schedule is considered to be the one with the lowest cost, which is measured by either simple deviations from specified due dates or total costs calculated in more sophisticated ways.
- *Maximize profit*: Given available equipment and other resources, the objective is to find the optimal schedule with the highest value of overall profit in a specified time horizon.

The complexity of the scheduling problems necessitates the development of effective schemes for organizing the large amount of information required to describe most scheduling applications. For instance, Zentner, Elkamel, Pekny, and Reklaitis (1998) proposed a high level language as a compact and context independent means of expressing a wide variety of process scheduling problems. On the other hand, as illustrated by Honkomp, Lombardo, Rosen, and Pekny (2000), many of the features of the process scheduling problems described above, such as resource sharing and inventory constraints, make these scheduling problems difficult to solve and present challenges to the regular use of scheduling technologies.

Note that in the literature on process scheduling, a number of earlier work addressed the traditional campaign operations, such as those presented by Wellons and Reklaitis (1991a,b) and Tsirukis, Papageorgaki, and Reklaitis (1993). As pointed out by Applequist et al. (1997), a key limitation of these campaign-based approaches arises from the restriction of the cyclic campaign operation which usually lead to lower equipment utilization and higher levels of inventories. In this review, we focus on the more general and flexible operational mode.

3. 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. Two of the earliest research contributions that employed this type of discrete time representation were presented by Bowman (1959) and Manne (1960) for jobshop 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 jobshop scheduling. More recently, the same concept was introduced to the chemical engineering literature for the general scheduling problem of a wide variety of chemical processes. Examples of work based on this approach include those presented by Kondili et al. (1993), Shah, Pantelides, and Sargent (1993), Pantelides (1993), Dedopoulos and Shah (1995), Pekny and Zentner (1993), Zentner, Pekny, Reklaitis, and Gupta (1994), Bassett, Pekny, and Reklaitis (1996) and Elkamel, Zentner, Pekny, and Reklaitis (1997).

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, which will be illustrated below with a general discrete-time formulation proposed by Kondili et al. (1993) and Shah et al. (1993) based on the STN representation.

One of the common components in scheduling problems involves the allocation of units to tasks. To model these assignments, binary variables W_{ijt} 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 I_j is the set of tasks that can be performed in unit (j), α_{ij} 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., $W_{ijt} = 1$), then no other task can start in the same unit until task (i) is finished after the duration of α_{ij}. Note that the latter constraint becomes trivially satisfied when task (i) does not start at time (T) (i.e., $W_{ijt} = 0$).

To account for batch-sizes and mass balances, continuous variables B_{ijt} are used to represent the amount of material which starts undergoing task (i) in unit (j) at time interval (T), and S_{st} 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:

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

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

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

where $V_{ij}^{\min}$ and $V_{ij}^{\max}$ are the minimum and maximum capacity of unit (j) for task (i), respectively; I_s^{p} and I_s^{c} the set of tasks that produce and consume state (s), respectively; ρ_{is}^{p} and ρ_{is}^{c} the fractions of state (s) produced and consumed by task (i), respectively; J_i the set of units suitable for task (i); α_{is} the processing time for state (s) by task (i); and C_s is the storage capacity limit for state (s). R_{st} is the amount of state (s) received from external sources at time interval (T). Variable D_{st} 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., $W_{ijt} = 1$), the batch-size is bounded by the minimum and maximum capacities of the involved unit ($V_{ij}^{\min} \leq B_{ijt} \leq V_{ij}^{\max}$). When the task does not take place (i.e., $W_{ijt} = 0$), the corresponding batch-size is zero (i.e., $B_{ijt} = 0$). 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).

The sequence-dependent changeover is another important element in many process scheduling problems. It can be incorporated by introducing cleaning tasks $i_{jff'}$ to model the changeover required for unit (j) to switch from family (f) to family (f'). Then this task must take place if a task for family (f) and a second task for family (f') are performed in unit (j) consecutively, which can be written as the following constraint:

$$\sum_{t=t_1+1}^{t_2-1} W_{i_{jff'},j,t} \geq \sum_{i \in I_j^{(f)}} W_{ijt_1} + \sum_{i \in I_j^{(f')}} W_{ijt_2} - \sum_{t=t_1+1}^{t_2-1} \sum_{i \in I_j} W_{ijt} - 1,$$
$$\forall j \in J,\ f \neq f' \in F,\ t_1 < t_2 \in T, \tag{6}$$

where f is the set of product families, and $I_j^{(f)}$ and $I_j^{(f')}$ are the set of tasks for family (f) and (f'), respectively. The above constraint is only meaningful when the first two sums on the right hand side are equal to one and the third one is zero. An alternative way is to account for the case that if a task for family (f) starts processing in unit j, no task for family (f') can start for a period of time required for the related cleaning operation after the end of the first task. This can be expressed as

$$\sum_{i' \in I_j^{(f')}} \sum_{t'=t+\alpha_{ij}}^{t+\alpha_{ij}+\tau_{jff'}-1} W_{i'jt'} \leq M(1 - W_{ijt}),$$
$$\forall j \in J,\ i \in I_j^{(f)},\ t \in T, \tag{7}$$

where $\tau_{jff'}$ is the duration of the cleaning operation required for unit (j) to switch from family (f) to family (f').

In a simple manner analogous to those shown above, discrete-time formulations can also model other relationships involved in various scheduling problems, such as product delivery at intermediate due dates, renewable resource considerations and a variety of schedule performance criteria. These resource constrained formulations are straightforwardly formulated models which usually lead to well-structured mathematical programming problems. However, they have two main limitations: the discrete approximation of time and the large size of resulting MILP problems.

Because of the continuous nature of time and the concept of discretization, the discrete-time formulations are by definition only approximations of the actual problem. Furthermore, one of the key issues in these approaches is the selection of the duration of the uniform time intervals, which always presents a tradeoff between the accuracy or quality of solution and the required computational efforts. If the chemical process of interest involves only fixed processing times, it is possible to model the process accurately, but it is required to use the greatest common factor of the processing times as the duration of each time interval, which usually leads to very large combinatorial problems for real-world applications that are difficult or even impossible to solve. If a coarser discretization scheme is used, the problem size may become more tractable, but there is an inevitable loss of model accuracy and it results by definition in suboptimal solutions. Furthermore, for operations with variable processing times such as continuous processes, which consume feeds and produce products continuously and in general can potentially run for a time period of any duration, the discrete-time approaches provide approximate descriptions of the actual processes, which can deviate substantially from the true solutions.

To reduce the difficulty in solving the large MILP problems resulting from the discrete-time models, a number of techniques have been proposed to improve the solution efficiency by exploiting the characteristics of the problem. These techniques include: (i) reformulation that reduce the gap between the optimal solution and its LP relaxation counterpart, for example, Sahinidis and Grossmann (1991b), Shah et al. (1993) and Yee and Shah (1998) reformulated the allocation and/or batch-sizing constraints based on variable aggregation/disaggregation; (ii) adding cut constraints which are redundant but reduce the region of integer infeasibility, such as those proposed by Dedopoulos and Shah (1995) and Yee and Shah (1998); (iii) intervening in the branch and bound solution procedure, for instance, Shah et al. (1993) developed ways to reduce the size of the relaxed LP and perform post analysis of the LP solution at each node of the branch and bound tree, Dedopoulos and Shah (1995) proposed techniques to fix variables to values implied during the branch and bound procedure; (iv) decomposition that divides a large and complex problem to smaller subproblems, for example, Bassett, Pekny, et al. (1996) proposed a number of time-based decomposition approaches and Elkamel et al. (1997) developed another algorithm consisting of both spatial decomposition and temporal decomposition.

4. Continuous-time approaches

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.

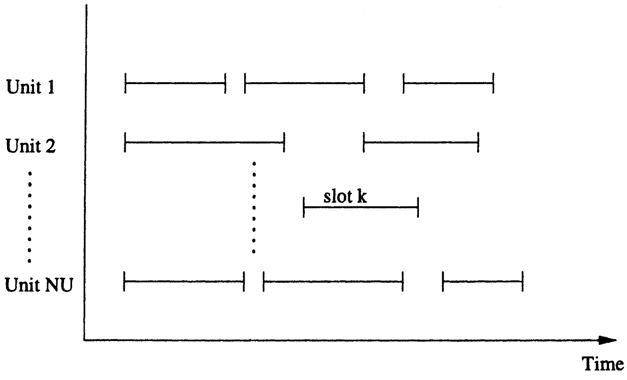

Fig. 4. Time slots defined for each unit.

4.1. 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. 4. 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).

To illustrate this type of models, let us consider the multistage flowshop problem with parallel units at each stage. The following key variables are defined:W_{ijkl}: binary, determines whether or not stage (l) of order (i) is assigned to the time slot (k) of unit (j); Tsi_{il}, Tei_{il}: continuous, the starting and completion times of stage (l) of order (i); Ts_{jk}: continuous, the starting time of time slot (k) of unit (j).

The allocation constraints are written as

$$\sum_{j\in(J_i\cap J_l)}\sum_{k\in K_j} W_{ijkl}=1,\quad \forall i\in I,\ l\in L_i, \tag{8}$$

$$\sum_{i\in I_j}\sum_{l\in(L_i\cap L_j)} W_{ijkl}\le 1,\quad \forall j\in J,\ k\in K_j. \tag{9}$$

In terms of timing, the following constraints are formulated to match the starting time of batches to the time slots and to correlate the starting time of an order at a stage with the ending time for the same order at the previous stage:

$$-M(1-W_{ijkl})\le(\mathrm{Tsi}_{il}-\mathrm{Ts}_{jk})\le M(1-W_{ijkl}),$$
$$\forall i\in I,\ J\in(J_i\cap J_l),\ k\in K_j,\ l\in L_i, \tag{10}$$

$$\mathrm{Tei}_{il}\le\mathrm{Tsi}_{i,(l+1)},\quad \forall i\in I,\ l\in L_i-\{l_i^l\}. \tag{11}$$

It should be noted that this type of model generally requires a large number of binary variables resulting from the introduction of variables W_{ijkl}, which are defined on four dimensions (i.e., the number of variables is equal to ($i \times j \times k \times l$). Furthermore, in all the time-slot based models, a predefined number of time slots is used and hence optimality cannot be guaranteed.

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).

For comparison with the slot-based approaches, let us consider the formulation of Méndez, Henning, and Cerdá (2001) for scheduling of multistage flowshop problems with parallel units, which was based on the notion of order predecessor. The key variables are defined as follows:

Y_{ij}: binary, to allocate order (i) to unit (j); $X_{ii'l}$: binary, defined only once for every pair of orders (i) and (i'), =1, if (i) precedes (i') at stage (l), = 0 otherwise; C_{il}: completion time of order (i) at stage (l).

The allocation constraints now take the form of

$$\sum_{j\in J_{il}} Y_{ij} = 1, \quad \forall i \in I,\ l \in L_i, \tag{12}$$

which ensures that exactly one unit is assigned to each order at a related stage.

For sequencing and timing purposes, the timings of different orders processed consecutively in the same unit are maintained through the following constraints:

$$\begin{aligned} C_{i'l} - \mathrm{pt}_{i'j} \ge {} & C_{il} + \mathrm{su}_{i'j} + \tau_{ii'j} - M(1 - X_{ii'l}) \\ & -M(2 - Y_{ij} - Y_{i'j}), \\ \forall i, i' \in I,\ & i' > i,\ l \in L_{ii'},\ j \in J_{ii'l}, \end{aligned} \tag{13}$$

$$\begin{aligned} & C_{il} - \mathrm{pt}_{ij} \ge C_{i'l} + \mathrm{su}_{ij} + \tau_{i'ij} - MX_{ii'l} - M(2 - Y_{ij} - Y_{i'j}), \\ & \forall i, i' \in I,\ i' > i,\ l \in L_{ii'},\ j \in J_{ii'l}. \end{aligned} \tag{14}$$

For each order, the timing variables for different stages and delivery are connected as follows:

$$C_{il} \ge \sum_{j\in J_{il}} Y_{ij}(\mathrm{Max}[\mathrm{ru}_j, \mathrm{ro}_i]) + \mathrm{su}_{ij} + \mathrm{pt}_{ij}, \quad \forall i \in I,\ l \in L_i, \tag{15}$$

$$C_{il} \le C_{i,l+1} - \sum_{j\in J_{i,l+1}} Y_{ij}\mathrm{pt}_{ij}, \quad \forall i \in I,\ l \in L_i - \{l_i^l\}, \tag{16}$$

$$C_{il_i^l} \le d_i, \quad \forall i \in I, \tag{17}$$

where ru_j is the earliest available time of unit (j); ro_i the earliest time at which order (i) can start; su_{ij} and pt_{ij} the setup time and processing time for order (i) in unit (j), respectively; d_i is the due date of order (i).

Table 1
Comparison between slot based and non-slot based formulations

Formulation	Slot based, Pinto and Grossmann (1995)	Non-slot based, Méndez, Henning, and Cerdá (2001)
Binary variable	**161**	**155**
Continuous variable	365	25
Constraints	449	477
Objective (maximize)	**6151**	**6269**
Nodes	851	201
CPU time (s)	58.87[a]	1.53[b]

[a] HP 9000-730.
[b] Pentium H PC (400 MHz).

Compared to the slot-based formulations, since this approach features continuous variables directly representing the task timings, it does not rely on time slots for task-unit allocation and hence can be more accurate and lead to better solutions. However, to model the sequencing of competing tasks in shared units, binary variables, such as $X_{ii'l}$ described above, and corresponding constraints are introduced to determine the relative order of tasks. Consequently, the size and complexity of the resulting model of this approach is usually comparable to that of the slot-based approach, both leading to large scale mathematical programming problems for real-world applications. As an illustrative example, Table 1 compares a slot based formulation and a non-slot based formulation applied to the scheduling of a multiproduct batch plant with 5 stages and 25 units that manufactures dyes. The formulations resulted in MILP models of similar sizes in terms of the number of binary variables, however, the formulation proposed by Méndez et al. (2001) led to better solution than the one found by Pinto and Grossmann (1995).

The continuous-time models that have been developed for sequential processes can be applied to scheduling problems with various features, including different intermediate storage policies, sequence-dependent change-overs, batch and continuous processes, intermediate due dates, renewable resource restrictions, and different objective functions such as minimization of order earliness and minimization of makespan.

4.2. 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.”

4.2.1. *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. We discuss the key components of their RTN formulation to illustrate this type of model.

The most important variables in this formulation include: T_k: continuous, timing of event (k); W_{ijk}: binary, whether or not task (i) starts at T_k in unit (j); $X_{ijkk'}$: binary, activated if task (i) starts at T_k in unit (j) and completes at $T_{k'}$.

The timings of events are required to be monotonically increasing.

$$0 = T_1 < T_2 < \cdots < T_K \leq H, \tag{18}$$

where H is the time Horizon.

The following allocation constraints are written to ensure that if task (i) starts in unit (j) at event time (k), it finishes at exactly one later event time:

$$W_{ijk} = \sum_{k' \geq k} X_{ijkk'}, \quad \forall i \in I,\ j \in J_i,\ k \in K. \tag{19}$$

The duration of a task, represented by variable t_{ijk}, is determined by the following timing constraint:

$$t_{ijk} = \sum_{k' > k} X_{ijkk'}(T_{k'} - T_k), \quad \forall i \in I,\ j \in J_i,\ k \in K. \tag{20}$$

Note that the above equation involves bilinear products of binary and continuous variables. Exact linearization techniques (Glover, 1975; Floudas, 1995) can be applied to transform them into linear forms at the expense of introducing additional variables and constraints.

This formulation leads to large scale MINLP problems. For certain classes of problems, for example, in the case of batch processes with simple objective function, the model can be linearized, but introduces a large number of additional variables and constraints.

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. There are two main differences between their formulation and those of Zhang (1995) and Mockus and Reklaitis (1997). First, they defined the durations of time slots, τ_k, as the main timing variables instead of the absolute times of the slot boundaries. Second, they introduced binary variables $y_{ikk'}$ to take the value of one if task (i) starts at time (k) and is active over slot (k') $\geq$ (k). Exact linearization techniques are also required to remove the nonlinearities arising from products of integer and continuous variables. For the solution of the resulting large scale MILP/MINLP problems, they developed a special branch and bound algorithm which branches on both the continuous τ_k variables and the binary variables.

More recently, Castro et al. (2001) proposed an RTN based MILP continuous-time formulation for the short-term scheduling of batch processes. They defined binary variables to determine the beginning and end of tasks at event points as well as continuous variables for timings of events and formulated the constraints, such as excess resource constraints, in a very similar way to that in the approach of Schilling and Pantelides (1996). Majozi and Zhu (2001) proposed an MILP continuous-time formulation for the short-term scheduling of batch processes based on a new process representation called State Sequence Network. They used time points to denote the use or production of states and introduced binary variables $y(s, p)$ associated with the usage of state (s) at time point (p). Their formulation leads to small MILP problems, but relies on the definition of effective states which are related to tasks and units. Lee et al. (2001) reported an STN based MILP formulation for batch and continuous processes, which introduced three sets of binary variables to account for the start, process, and end events of each task. Burkard et al. (2002) developed an STN based MILP formulation for the makespan minimization problem for batch processes and discussed the choice of the objective function and additional constraints. Wang and Guignard (2002) presented an STN based MILP formulation for batch process scheduling problems, which proposed the definition of events associated with inventory changes to reduce the total number of events required to model a schedule.

The global event based continuous-time models can incorporate a wide variety of considerations in process scheduling, such as intermediate storage, change-over, batch and continuous operational modes, due dates, renewable resources, and various objective functions.

As pointed out by Zhang and Sargent (1996), for all the global event based continuous-time models discussed above, an important issue (in addition to the required linearizations) is the estimation and adjustment of the number of events/time slots/time points. An underestimation may lead to suboptimal solutions or even infeasible problems, while an overestimation results in unnecessarily large problems, which increase even more the difficulty of the solution. Despite its significance, there has been relatively little attention on this issue in the literature. The only exceptions are the work presented by Schilling (1997) and very recently by Castro et al. (2001). They proposed an iterative procedure in which the model begins with a small number of events and then the number is gradually increased until no improvement can be achieved. However, as reported by Castro et al. (2001), in some cases, the solution may improve only after the addition of more than one event, which creates difficulty for the establishment of a stopping criterion that can guarantee the optimality of the solution.

4.2.2. Unit-specific event based models

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 representing the beginning of a task or utilization of the unit. The basic idea is illustrated in Fig. 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.

Two sets of binary variables are defined: wv(i, n) to determine whether or not task (i) starts at event point (n); yv(j, n) to determine whether or not unit (j) starts being utilized at event point (n). They are connected through the following allocation constraint:

$$\sum_{i \in I_j} \mathrm{wv}(i, n) = \mathrm{yv}(j, n), \quad \forall j \in J,\ n \in N. \tag{21}$$

These constraints express that in each unit (j) and at an event point (n) at most one of the tasks that can be performed in this unit (i.e., $i \in I_j$,) should take place. Note that if a task can be performed in multiple units, it is split into multiple tasks with each one performed in a different unit. This will increase the number of wv(i, n) binary variables and in the worst case where every task can take place in every unit, the total number of tasks after splitting is equal to the number of the original tasks times the number of units.

Continuous variable $B(i, j, n)$ represents the batch-size of task (i) in unit (j) at event point (n) and it is correlated with binary variable wv(i, n) through the following capacity

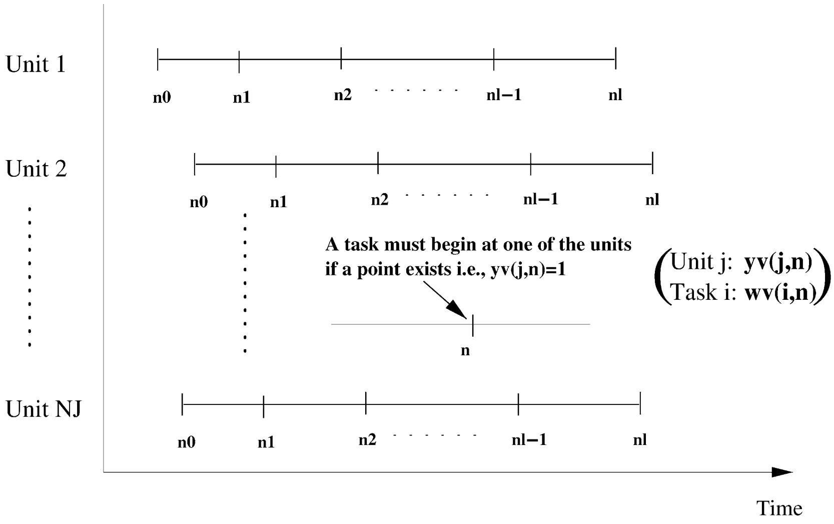

Fig. 5. Event points defined for each unit.

constraint:

$$V_{ij}^{\min}\text{wv}(i,n) \le B(i,j,n) \le V_{ij}^{\max}\text{wv}(i,n),\quad \forall i \in I,\ j \in J_i,\ n \in N. \tag{22}$$

ST(*s*, *n*) and *D*(*s*, *n*) are the amounts of state (*s*) stored and delivered at event point (*n*), respectively; they are used in the following material balance constraint:

$$\begin{aligned}\text{ST}(s,n) &= \text{ST}(s,(n-1)) - D(s,n) \\ &\quad + \sum_{i\in I_s^p} \rho_{is}^{\text{p}} \sum_{j\in J_i} B(i,j,(n-1)) \\ &\quad - \sum_{i\in I_s^c} \rho_{is}^{\text{c}} \sum_{j\in J_i} B(i,j,n), \quad \forall s \in S,\ n \in N.\end{aligned} \tag{23}$$

The above constraint is written for batch tasks, while similar constraints can be written for continuous tasks which takes into account the different nature of the continuous operation mode (see Ierapetritou & Floudas, 1998b). It should be pointed out that the amount of state (*s*) at an event point (*n*), represented by st(*s*, *n*), generally does not correspond to one well-defined time instance or time period due to the fact that the state can be consumed or produced by different tasks that take place in different units with different time axis. When there is a storage limit for the state, an upper bound can be imposed on the ST(*s*, *n*) variable as an approximate way to model the storage restrictions. A rigorous way is to introduce storage tasks and storage units with certain capacity ranges (see Ierapetritou & Floudas, 1998b; Lin & Floudas, 2001).

The last two sets of main continuous variables $T^{\text{s}}(i, j, n)$ and $T^{\text{f}}(i, j, n)$ represent the starting and ending times of task (*i*) in unit (*j*) at event point (*n*). The various relationships among the timings of tasks are formulated by two types of timing constraints: duration constraints and sequence constraints. The duration of a task is expressed as a linear function of the batch-size as follows:

$$T^{\text{f}}(i,j,n) = T^{\text{s}}(i,j,n) + \alpha_{ij}\text{wv}(i,n) + \beta_{ij}B(i,j,n),\quad \forall i \in I,\ j \in J_i,\ n \in N. \tag{24}$$

This general form of variable processing times is able to deal with a wide variety of processes. For tasks with fixed processing times, α_{ij} corresponds to the processing time of the task and β_{ij} is zero. While for tasks operating in the continuous mode, α_{ij} is zero and β_{ij} is the inverse of the processing rate. When a range is given for the processing rate, the related constraints are then written as

$$\begin{aligned}&R_{ij}^{\min}[T^{\text{f}}(i,j,n) - T^{\text{s}}(i,j,n)] \\ &\quad \le B(i,j,n) \le R_{ij}^{\max}[T^{\text{f}}(i,j,n) - T^{\text{s}}(i,j,n)], \\ &\forall i \in I,\ j \in J_i,\ n \in N,\end{aligned} \tag{25}$$

$$T^{\text{f}}(i,j,n) - T^{\text{s}}(i,j,n) \le H\text{wv}(i,n),\quad \forall i \in I,\ j \in J_i,\ n \in N, \tag{26}$$

where $R_{ij}^{\min}$ and $R_{ij}^{\max}$ are the minimum and maximum processing rates of unit (*j*) when performing task (*i*), respectively.

A very important element of the formulation is the following three sets of special sequencing constraints:

- *Same task in the same unit*:

$$T^{\text{s}}(i,j,(n+1)) \ge T^{\text{f}}(i,j,n),\quad \forall i \in I,\ j \in J_i,\ n \in N,\ n \ne n_{\text{last}}. \tag{27}$$

  These constraints state that task (*i*) starting in unit (*j*) at event point (*n* + 1) should start after the end of the same task performed in the same unit which has already started at event point (*n*).
- *Different tasks in the same unit*: The following constraints establish the relationship between the starting time of a task (*i*) at event point (*n* + 1) and the ending time of task (*i*′) at event point (*n*) when these tasks take place in the same unit (*j*).

$$\begin{aligned}T^{\text{s}}(i,j,(n+1)) &\ge T^{\text{f}}(i',j,n) + \tau_{ji'i}\text{wv}(i',n) \\ &\quad - H(1-\text{wv}(i',n)), \\ \forall j \in J,\ i \in I_j,&\ i' \in I_j,\ i \ne i',\ n \in N,\ n \ne n_{\text{last}}.\end{aligned} \tag{28}$$

  If wv(*i*′, *n*) = 1 which means that task (*i*′) takes place in unit (*j*) at event point (*n*), then the last term of constraint (28) becomes zero forcing task (*i*) in unit (*j*) at event point (*n* + 1) to start after the ending time of task (*i*′) in unit (*j*) at event point (*n*) plus the required clean-up time; otherwise the right hand side of constraint (28) becomes negative and the constraint is trivially satisfied. Note that the sequence-dependent changeover is directly incorporated in this constraint. It should also be pointed out that this constraint actually imposes a lower bound not only on the starting time of task (*i*) at event point (*n* + 1) but also on the starting times of task (*i*) at the subsequent event points (*n* + 2), (*n* + 3), etc. because of the monotonically increasing relationships among the timings of the same task in the same unit at consecutive event points established by Constraint (27). In other words, if two tasks take place in the same unit consecutively, but at two event points with an idle event point in between, the requirement on their timings is also enforced.
- *Different tasks in different units*: The following constraints are written for different tasks (*i*, *i*′) that are performed in different units (*j*, *j*′) but take place consecutively according to the production recipe due to material connections:

$$T^{\text{s}}(i,j,(n+1)) \ge T^{\text{f}}(i',j',n) - H(1-\text{wv}(i',n)),\quad \forall j, j' \in J,\ i \in I_j,\ i' \in I_{j'},\ i \ne i',\ n \in N,\ n \ne n_{\text{last}}. \tag{29}$$

  If task (*i*′) takes place in unit (*j*′) at event point (*n*) (i.e., wv(*i*′, *n*) = 1), then we have $T^{\text{s}}(i, j, (n+1)) \ge T^{\text{f}}(i', j', n)$ and hence task (*i*) in unit (*j*) has to start after the end of task (*i*′) in unit (*j*′). Similar to Constraint (28), this constraint also establishes the relationships between tasks that are assigned to non-consecutive event points.

The zero-wait condition can also be incorporated by adding the following constraint for tasks (i, i') that take place consecutively without delay due to storage restrictions on the intermediate material.

$$T(i, j, (n+1)) \leq T^{\mathrm{f}}(i', j', n) + H(2 - \mathrm{wv}(i, (n+1)) - \mathrm{wv}(i', n)),$$
$$\forall i, i' \in I,\ j \in J_i,\ j' \in J_{i'},\ n \in N,\ n \neq n_{\mathrm{last}}. \tag{30}$$

With constraints (28) or (29), this constraint enforce that task (i) in unit (j) at event point $(n+1)$ starts immediately after the end of task (i') in unit (j') at event point (n) if both of them are activated.

The above duration and sequencing constraints model the sequencing and timing relationships efficiently, avoiding the introduction of any additional variables such as $X_{ijkk'}$ in the global event based models discussed in the previous section. This contributes to the significant further reduction of the number of binary variables.

Demands with intermediate due dates can also be incorporated in this formulation. This is achieved by linking demands to event points and enforcing the following constraints on the amounts and timings of product deliveries:

$$D(s, n) + \mathrm{SL}(s, n) = \mathrm{da}_{sn}, \quad \forall s \in S,\ n \in N, \tag{31}$$

$$T^{\mathrm{s}}(i, j, n) \leq \mathrm{dd}_{sn}, \quad \forall s \in S, i \in I_s^{\mathrm{p}},\ j \in J_i, \tag{32}$$

where da_{sn} and dd_{sn} are the amount and due date of the demand for state (s) at event point (n). Note that only the tasks that produce the involved state are considered in Constraint (32). $\mathrm{SL}(s, n)$ are slack variables introduced to give more flexibility to the model in handling partial fulfillment of demands. Under feasible conditions, some or all of these variables can be fixed to zero to ensure that some or all of the demands within the time horizon are met.

Similar to the global event based models, this formulation is also faced with the important issue of the determination of the number of event points. The general procedure is to start with a small number and iteratively increase it until no improvement of the objective function can be achieved (Ierapetritou & Floudas, 1998a). The possibility that it requires the addition of more than one event point to improve the solution for this formulation is much smaller than that for the global event based models, due to the more efficient utilization of event points and the smaller number of event points required to model a process.

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. This can be demonstrated by comparing the different approaches applied to a small example which involves the process described by the STN in Fig. 2 or the RTN in Fig. 3 and requires variable processing times. As shown in Table 2, the discrete-time approach is an approximation of the actual process and by definition leads to suboptimal solutions that can deviate from the optimal solution substantially. Furthermore, the size of the resulting model and the required solution time explode exponentially as the number of time intervals increases to improve the degree of accuracy. When the number of time intervals is increased from 8 (corresponding to a discretization interval of 1 h) to 32 (corresponding to a discretization interval of 0.25 h), the discrete-time formulation attains better approximation and better solution, which improves from 620.2 to 1195.3. Note that this is still suboptimal compared to the best solution of 1498.2 obtained through a unit-specific event based continuous-time formulation. Also, the size of the resulted model grows significantly, which is mainly reflected by the number of binary variables that increases from 38 to 591, and the required solution time explodes from less than a second to more than 100,000 s without solving to optimality. In contrast, continuous-time approaches lead to more accurate models of smaller sizes. The global event based formulation proposed by Zhang (1995) led to an MILP model with 147 binary variables, which can be solved in reasonable time and achieves a much better objective value of 1497.7. The unit-specific event based approach proposed by Ierapetritou and Floudas (1998a, 2001) used a smaller number of events and further reduces the size of the resulting model to 40

Table 2
Comparison between discrete-time models and global event based, unit-specific event based continuous-time models

Model	Discrete-time models			Continuous-time models	
				Global event based, Zhang (1995)	Unit-specific event based, Ierapetritou and Floudas (1998a, 2001)
Events/time intervals	8	16	32	7	5
Binary variable	**38**	**171**	**591**	**147**	**40**
Continuous variable	743	2386	8590	497	260
Constraints	1567	5135	18415	741	374
Objective (profit)	**620.2**	**940.5**	**1195.3**	**1497.7**	**1498.2**
Nodes	15	5123	~500,000	9575	51
CPU time (s)	0.29[a]	58[a]	~100,000[a]	1027.5[b]	0.28[a]

[a] HP-C160.
[b] SunSparc10/41.

Table 3
Comparison between sequential process oriented models and general process oriented models (example 3 in Karimi and McDonald (1997))

Continuous-time formulation	Slot based, Karimi and McDonald (1997)	Unit-specific event based, Ierapetritou, Hené and Floudas (1999)
Binary variable	**152**	**140**
Continuous variable	1378	1277
Constraints	1417	5927
CPU time (s)	320[a]	35.6[b]
Nodes	2563	802
Cost	**350257**	**350216**
LP relaxation	302934	259369

[a] IBMRS-6000.
[b] HP-C160.

binary variables. The model led to an objective value of 1498.2 and was solved with much less computational effort. The advantages have rendered this approach the capability of addressing large scale industrial applications in medium-range production scheduling (e.g., Lin, Floudas, Modi, and Juhasz, 2002).

It should be pointed out that the continuous-time models for network-represented processes have been developed for general chemical processing systems and can be employed to address scheduling problems of sequential processes as well. Furthermore, in this case, the unit-specific event based model also performs better than other existing approaches in terms of the size of the resulting model and the quality of the optimal schedule, which can be illustrated in Table 3 for a semi-continuous process.

5. Computational studies and applications

The various scheduling formulations have been employed to study a considerable number of problems. Table 4 summarizes some of the largest model sizes that have been reported in the literature.

With the substantial advances in the modeling and solution of scheduling formulations, they have been applied to a wide variety of real-world problems. Some examples of notable applications are presented below. Sahinidis and Grossmann (1991a) proposed a time slot based MINLP model for the cyclic scheduling of multiproduct plants with continuous parallel lines and applied it to the scheduling of polymer production in three parallel plants of a large chemical company. Shah et al. (1993) described a case study on the scheduling of a hydrolubes plant with adapted industrial data using the STN based discrete-time approach. Wilkinson, Cortier, Shah, and Pantelides (1996) addressed a large scale production and distribution scheduling problem in which three multipurpose production facilities in different countries supply a large portfolio of fast moving consumer goods to the European market. They proposed a detailed formulation that considered all three plants simultaneously. Due to the very large size of the resulting problem, they generated an approximate

Table 4
Examples of largest problems reported in the literature

Work	Process feature	Time representation	Resulting model	Solution approach	Model size (binary variables, continuous variables, constraints)	Objective	CPU time
Shah et al. (1993)	Network (STN) 12 states/6 tasks/6 units/FIS, NIS	Discrete	MILP STN-based	Modified branch and bound	2316, 3855, 3376	Maximize added value	48 min[a]
Pinto and Grossmann (1994)	Sequential, continuous, cyclic 8 products/3 stages/3 units	Continuous slot-based	MINLP	GBD/OA/AP	448, 1970, 3002	Maximize profit	49 min[b]
Pinto and Grossmann (1995)	Sequential 50 orders/5 stages/25 units	Continuous slot-based	MILP	Decomposition	1050, 47917, 48843	Minimize earliness	4.2 h[c]
Schilling and Pantelides (1996b)	Network (RTN), continuous FIS, 15 products/11 units	Continuous	MILP	Modified branch and bound	1042, 2746, 4981	Maximize net profit	57 min[d]
Zhang and Sargent (1998)	Network (RTN), continuous FIS, 15 products/11 units	Continuous	MILP	CPLEX	1318, 3237, 4801	Maximize net profit	18 min[e]
Ierapetritou and Floudas (1998b)	Network (STN), batch and continuous FIS, 28 products/13 units	Continuous	MILP	GAMS/CPLEX MINOPT/CPLEX	2375, 29384, 51000	Minimize makespan	2.7 h[f]

[a] SUN SparcStation IPX.
[b] HP 9000-750.
[c] HP 9000-730.
[d] Six parallel processors.
[e] Sun-Sparc 10/41.
[f] HP-C160.

formulation by aggregating constraints, whose solution gave a tight upper bound on the production capacity and facilitated the decomposition of the original problem into small sub-problems, each involving a single plant.

Zhang (1995), Schilling and Pantelides (1996b), Ierapetritou and Floudas (1998b) considered the scheduling of an industrial fast-moving consumer goods manufacturing plant involving batch and continuous processes. Continuous-time formulations were proposed using either global events (Zhang, 1995; Schilling & Pantelides, 1996b) or unit-specific events (Ierapetritou & Floudas, 1998b). Ierapetritou et al. (1999) extended the continuous-time formulation in (Ierapetritou & Floudas, 1998a,b) to deal with intermediate due dates and addressed a variety of problems, including the short-term scheduling of a single-stage multiproduct facility with multiple semicontinuous processors. Jain and Grossmann (1999) investigated the resource-constrained scheduling of testing tasks for new product development in pharmaceutical and agrochemical industries. Slot-based MILP models were proposed to solve the scheduling problem. Pinto, Joly, and Moro (2000) discussed planning and scheduling applications for refinery operations. Both continuous and discrete time formulations were developed for the scheduling of refinery production and distribution. Méndez and Cerdá (2000a) addressed the short-term scheduling of a two-stage multiproduct batch plant which delivered intermediate products to nearby end-product facilities and proposed a continuous-time MILP model. Georgiadis, Papageorgiou, and Macchietto (2000) considered the short-term cleaning scheduling in a special class of heat exchanger networks involving decaying equipment performance due to milk fouling. Discrete-time approaches were employed to formulate an MINLP model incorporating general fouling profiles, which is then linearized and solved as an MILP problem. Yi, Suh, Lee, and Lee (2000) studied the production scheduling of a polybutene process featuring the requirement of product quality check in intermediate storage tanks and developed a discrete-time MILP model.

Glismann and Gruhn (2001) developed an approach to integrate the short-term scheduling of multiproduct blending facilities and nonlinear recipe optimization. An RTN based discrete-time MILP model was formulated for the scheduling problem. Harjunkoski and Grossmann (2001) presented a decomposition algorithm for the short-term scheduling of large scheduling problems in the steel making industry. The original problem is disaggregated into subproblems based on product groups characteristic of steel making and a sequence of smaller MILP problems are solved followed by an aggregating MILP. Lin, Chajakis, and Floudas (2003) investigated the scheduling of marine vessels for tanker lightering in the crude oil supply chain. A novel continuous-time MILP formulation was developed based on the concept of event points, in which a task consists of a sequence of operations performed by a vessel during the lightering process. Lamba and Karimi (2002b) developed a two-step decomposition algorithm for the short-term scheduling of a single-stage multiproduct facility with multiple semicontinuous production lines. The algorithm is based on item combinations and is applied to an industrial problem from a detergent plant. Castro, Matos, and Barbosa-Póvoa (2002) addressed the scheduling of a batch digester cooking system of an industrial acid sulphite pulp mill constrained by steam availability. A discrete time RTN based model featuring the most relevant steam-sharing alternatives was developed and the required process data were obtained with a dynamic model of the heating system. Lin, Floudas, Modi, and Juhasz (2002) presented a systematic framework for the medium-range production scheduling of a large industrial polymer multiproduct plant, which uses a rolling horizon decomposition approach coupled with the unit-specific event based continuous-time scheduling formulation.

6. Integrated synthesis, design and scheduling of multiprod-uct/multipurpose plants

The inherent operational flexibility of multiproduct/multipurpose plants gives rise to considerable complexity in the design and synthesis of such plants. In many cases, scheduling strategies are not incorporated or integrated very well, which may lead to over-design or under-design. In order to ensure that any resource incorporated in the design can be used as efficiently as possible, detailed considerations of plant scheduling must be taken into account at the design stage. Therefore, it is important to consider design, synthesis and scheduling simultaneously.

There have been a number of publications in the area of design and operation of multiproduct/multipurpose plants. Sparrow, Forder, and Rippin (1975) and Grossmann and Sargent (1979) addressed the optimal design problem of sequential multiproduct batch plants taking into account scheduling issues by assuming campaigns of single products and including simple aggregated timing constraints. The resulting MINLP problems were solved using heuristics (Sparrow et al., 1975) or branch and bound techniques (Sparrow et al., 1975; Grossmann & Sargent, 1979).

On this basis, Suhami and Mah (1982) studied the optimal design of multipurpose batch plants focusing on a restricted form of the problem as the “unique unit-to-task assignment” case. The resulting MINLP problem was solved using an iterative procedure of solving NLP relaxations and adding constraints that corrected integer infeasibilities.

Vaselenak, Grossmann, and Westerberg (1987) also investigated the design and scheduling of multipurpose batch plants. They proposed a superstructure representation for products grouping and formulated an MINLP formulation, which was also solved as a sequence of NLP programs. Birewar and Grossmann (1989) proposed MINLP formulations for the design and scheduling of sequential multi-product batch plants which considered mixed product campaigns. Birewar and Grossmann (1990) extended them to address synthesis, sizing and scheduling of such plants simultaneously and used the AP/OA/ER algorithm implemented in DICOPT++ to solved the resulting MINLP problems.

Papageorgaki and Reklaitis (1990a,b, 1993) addressed the optimal design and retrofit of multipurpose batch plants. They pointed out that previous formulations omitted key aspects of the general multipurpose plant, such as alternative assignments of different equipment items to each product task and sharing of the units of the same equipment type among multiple tasks of the same or different products. They proposed formulations in which flexible unit-to-task allocations and non-identical parallel units are considered. Problem-specific decomposition strategies that iterated between an MILP or MINLP master problem and an NLP or MINLP upper bound subproblem were proposed to solve the resulting MINLP problems.

Crooks and Macchietto (1992) and Crooks, Kuriyan, and Macchietto (1992) integrated the synthesis, design and scheduling of general batch plants. They incorporated scheduling considerations by applying the STN framework and the discrete-time approach proposed by Kondili et al. (1993).

Voudouris and Grossmann (1993, 1996) presented MILP formulations for the optimal design and scheduling of sequential multiproduct and multipurpose batch plants under a number of assumptions such as discrete equipment sizes. Barbosa-Póvoa and Macchietto (1994) presented a detailed formulation of multipurpose batch plant design and retrofit based on the STN description and the discrete-time scheduling model proposed by Kondili et al. (1993). The resulting MILP problem was solved using a branch and bound method. Realff, Shah, and Pantelides (1996) considered the design problem for pipeless batch plants with mobile vessels and incorporated the STN-based discrete-time scheduling approach proposed by Kondili et al. (1993). A decomposition procedure is proposed to solve the resulting large MILP problems. Barbosa-Póvoa and Pantelides (1997) solved the multipurpose batch plant design problem using the RTN-based discrete-time scheduling model proposed by Pantelides (1993), which also resulted in MILP problems.

In recent years, continuous-time scheduling approaches have also been incorporated into the design problem for multiproduct/multipurpose plants. Xia and Macchietto (1997) presented a formulation based on the variable event time scheduling model of Zhang and Sargent (1996, 1998) and Zhang (1995). A stochastic method is used to solve the resulting nonconvex MINLP problems directly, instead of introducing a large number of auxiliary variables and constraints to reduce the MINLP into an MILP. Lin and Floudas (2001) extended the continuous-time scheduling formulation proposed by Ierapetritou and Floudas (1998a,b, 2001), Ierapetritou et al. (1999) to address the problem of integrated design, synthesis and scheduling of multipurpose batch plants. They studied both linear and nonlinear cases, which resulted in MILP and MINLP problems, respectively. The MILP problems were solved with an LP-based branch and bound method. The nonconvex MINLP problems were solved with MINOPT (Schweiger and Floudas, 1997) and global optimal solutions can be obtained for a class of problems with special structures.

7. Scheduling under uncertainty

Most of the scheduling models for chemical processes assume that all problem data are certain, that is, they are of constant known values and they are called deterministic models. However, uncertainty is prevalent in the context of scheduling in reality. The most common sources of uncertainty include: (i) process or model parameters, such as processing time and equipment availability; and (ii) environmental data, such as demand amount and/or due date, and price/cost of product/raw materials. It can be shown that a schedule generated by a deterministic model based on nominal values of the parameters may be infeasible upon realization of the uncertain parameters. It is thus very important to take into account uncertainty during the course of scheduling in order to improve the schedule quality.

Although there has been a substantial amount of work to address the problem of design and operation of batch plants under uncertainty (e.g., Shah and Pantelides, 1992; Subrahmanyam, Pekny, & Reklaitis, 1994; Ierapetritou & Pistikopoulos, 1996; Harding & Floudas, 1997; Petkov & Maranas, 1997), the issue of robustness in scheduling under uncertainty has received relatively less attention.

Existing approaches in the literature to deal with this problem can be divided into two groups: reactive scheduling and stochastic scheduling, and are presented below.

7.1. Reactive scheduling

The first approach, called reactive scheduling, handles uncertainty by adjusting a schedule upon realization of the uncertain parameters or occurrence of unexpected events. The original schedule is usually obtained a priori in a deterministic manner and reactive scheduling is performed either at or right before the execution of scheduled operations. Therefore, reactive scheduling systems are required to be able to generate updated schedules relatively quickly. It is not desirable to do full-scale rescheduling for every unexpected event and usually heuristic approaches are developed to achieve the purpose of schedule modifications.

One of the earliest efforts in reactive scheduling was reported by Cott and Macchietto (1989), which was a part of a larger computer aided production management system for batch processes. They considered fluctuations of processing times and used a shifting algorithm to modify the starting times of processing steps of a batch by the maximum deviation between the expected and actual processing times of all related processing steps. Kanakamedala, Reklaitis, and Venkatasubramanian (1994) considered deviations in processing times and unit availabilities in multipurpose batch plants. They developed a least impact heuristic search approach for schedule modification that allowed time shifting and unit replacement. Huercio, Espuna, and Puigjaner (1995) and Sanmarti, Huercio, Espuna, and Puigjaner (1996) proposed reactive scheduling techniques to deal with variations in task processing times and equipment availability. They used heuristic equipment selection rules for modification of task starting times and reassignment of alternative units.

Rodrigues, Gimeno, Passos, and Campos (1996) also considered uncertain processing times and proposed a reactive scheduling technique based on a modified batch-oriented MILP model according to the discrete-time STN formulation proposed by Kondili et al. (1993). A rolling horizon approach is utilized to determine operation starting times with lookahead characteristics taking into account possible violations of future due dates. Honkomp, Mockus, and Reklaitis (1999) proposed a reactive scheduling framework for processing time variations and equipment breakdown by coupling a deterministic schedule optimizer with a simulator incorporating stochastic events. A number of rescheduling strategies are proposed based on the discrete-time MILP scheduling model with the objective of minimizing the task starting time deviations from where they were originally scheduled. Vin and Ierapetritou (2000) considered two kinds of disturbances in multiproduct batch plants: machine breakdown and rush order arrival. They applied the continuous-time scheduling formulation proposed by Ierapetritou and Floudas (1998a, 2001), Ierapetritou et al. (1999) and reduced the computational effort required for the solution of the resulting MILP problems by fixing binary variables involved in the period before an unexpected event occurs. Roslöf, Harjunkoski, Bjorkqvist, Karlsson, and Westerlund (2001) developed an MILP based heuristic algorithm that can be used to improve an existing schedule or to reschedule jobs in the case of changed operational parameters by iteratively releasing a set of jobs in an original schedule and optimally reallocating them.

7.2. Stochastic scheduling

A second approach, called stochastic scheduling, takes into account the uncertainty information at the original scheduling stage and its objective is to create optimal and reliable schedules in the presence of uncertainty. The consideration of uncertainty transforms the problem from a deterministic one, where standard methods of mathematical programming can be applied, to a stochastic problem where special techniques are required.

For the well-studied flowshop problem, considerable work has been done in operations research, focusing on uncertain processing times. A variety of rules or sufficient conditions for optimal solutions were proposed or identified to facilitate the development of efficient scheduling algorithms for specific classes of problems. Examples of such methods can be found in Hamada and Glazebrook (1993) and Kamburowski (1999, 2000).

In the chemical engineering literature, an approach based on the framework of scenarios attempts to forecast and account for all possible future outcomes through the use of a number of scenarios, using either discrete probability distributions or the discretization of continuous probability distribution functions. The expectation of a certain performance criterion, such as the expected makespan, is optimized with respect to the scheduling decision variables. Such methods provide a straightforward way to implicitly incorporate uncertainty. However, they inevitably enlarge the size of the problem significantly as the number of scenarios increases exponentially with the number of uncertain parameters. This main drawback limits the application of these methods to solve practical problems with a large number of uncertain parameters.

Bassett, Pekny, and Reklaitis (1997) presented a framework to take into account process uncertainties in processing time fluctuations, equipment reliability/availability, process yields, demands, and manpower changes. They used Monte Carlo sampling to generate random instances, determined a schedule for each instance, generated distribution of aggregated properties to infer operating policies. However, a specific robust schedule is not determined. Ierapetritou and Pistikopoulos (1996b) addressed the scheduling of single-stage and multistage multiproduct continuous plants with single production line at each stage when uncertainty in product demands is involved. They used Gaussian quadrature integration to evaluate the expected profit and formulated MILP models for the stochastic scheduling problem. Vin and Ierapetritou (2001) considered demand uncertainty for the short-term scheduling of general multiproduct and multipurpose batch plants based on the continuous-time MILP formulation proposed by Ierapetritou and Floudas (1998a). They introduced several metrics to evaluate the robustness of a schedule and proposed a multiperiod programming model using extreme points of the demand range as scenarios to generate a single sequence of tasks with the minimal average makespan over all scenarios.

Balasubramanian and Grossmann (2002) proposed a multiperiod MILP model for scheduling multistage flowshop plants with uncertain processing times described by discrete or continuous (using discretization schemes) probability distributions. The objective is to minimize expected makespan and a special branch and bound algorithm was used based on lower bounding by an aggregated probability model.

Sanmarti, Espuna, and Puigjaner (1997) presented a different approach for the scheduling of production and maintenance tasks in multipurpose batch plants in the face of equipment failure uncertainty. They computed a reliability index for each unit and for each scheduled task and formulated a nonconvex MINLP model to maximize the overall schedule reliability. Because of the significant difficulty in the rigorous solution of the resulting problem, a heuristic method was developed to find solutions that improve the robustness of an existing schedule.

There have also been attempts to transform a stochastic model to direct deterministic equivalent representation. This framework circumvents any need for explicit or implicit discretization or sampling of the uncertain data, avoiding undesirable increase of the problem size, and thus renders the potential capability of handling problems with a large number of uncertain parameters. Orçun, Altinel, and Hortaçsu (1996) considered uncertain processing times in batch processes and employed chance constraints to account for the risk of violation of timing constraints under certain conditions such as uniform distribution functions. Recently, Lin, Janak, and Floudas (2004) proposed a new robust optimization framework for scheduling general batch processes under uncertainty using a continuous-time MILP formulation. The underlying framework is based on a Robust Optimization methodology, which when applied to MILP problems produces “robust” solutions which are in a sense immune against uncertainties in both the coefficients and right-hand-side parameters of the inequality constraints. The approach can be applied to generate reliable schedules in the presence of uncertainty in processing times, product demands, and prices of materials and to gain insights on the tradeoffs between conflicting objectives.

8. Conclusions and perspectives

In this paper, we present an overview of the developments in the scheduling of multiproduct, multipurpose batch and continuous processes. Existing approaches were classified based on the time representation and important characteristics of chemical processes that pose challenges to the scheduling problem are discussed. In addition to the discrete-time approaches, various continuous-time models have been proposed in the literature and their strengths and limitations are examined. Computational studies and applications are presented and the integrated scheduling, design and synthesis, and scheduling under uncertainty are also reviewed.

It is apparent that significant advances have been made in the area of scheduling of chemical processes in the past decade. Further research work is needed to address classes of important large-scale industrial applications. More specifically, future research efforts should aim at addressing (a) the development of mathematical models and algorithms that reduce and even close the integrality gap for medium and large scale short-term scheduling applications; (b) medium-term scheduling of batch and continuous processes; (c) the multisite production and distribution scheduling; (d) the uncertainty in processing times, prices, changes in product demands, and equipment failure/breakdown; (e) the scheduling of manufacturing operations in the semiconductor industry in the presence of multiple reentrant flows; and (f) the integration of scheduling with design, synthesis, control and planning.

Acknowledgements

The authors gratefully acknowledge support from the National Science Foundation, and Atofina Chemicals, Inc.

References

Applequist, G., Samikoglu, O., Pekny, J., & Reklaitis, G. (1997). Issues in the use, design and evolution of process scheduling and planning systems. *ISA Transactions, 36*, 81–121.

Balasubramanian, J., & Grossmann, I. E. (2002). A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing times. *Computers and Chemical Engineering, 26*, 41–57.

Barbosa-Póvoa, A., & Macchietto, S. (1994). Detailed design of multipurpose batch plants. *Computers and Chemical Engineering, 18*, 1013–1042.

Barbosa-Póvoa, A.P.F.D., & Pantelides, C. C. (1997). Design of multipurpose plants using the resource-task network unified framework. *Computers and Chemical Engineering, 21*, S703–S708.

Bassett, M. H., Dave, P., Doyle III, F. J., Kudva, G. K., Pekny, J. F., Reklaitis, G. V., Subrahmanyam, S., Miller, D. L., & Zentner, M. G. (1996a). Perspectives on model based integration of process operations. *Computers and Chemical Engineering, 20*, 821–844.

Bassett, M. H., Pekny, J. F., & Reklaitis, G. V. (1996b). Decomposition techniques for the solution of large-scale scheduling probems. *AIChE Journal, 42*, 3373–3387.

Bassett, M. H., Pekny, J. F., & Reklaitis, G. V. (1997). Using detailed scheduling to obtain realistic operating policies for a batch processing facility. *Industrial and Engineering Chemistry Research, 36*, 1717–1726.

Birewar, D. B., & Grossmann, I. E. (1989). Incorporating scheduling in the optimal design of multiproduct batch plants. *Computers and Chemical Engineering, 13*, 141–161.

Birewar, D. B., & Grossmann, I. E. (1990). Simultaneous synthesis, sizing, and scheduling of multiproduct batch plants. *Industrial and Engineering Chemistry Research, 29*, 2242–2251.

Bok, J., & Park, S. (1998). Continuous-time modeling for short-term scheduling of multipurpose pipeless plants. *Industrial and Engineering Chemistry Research, 37*, 3652–3659.

Bowman, E. H. (1959). The schedule-sequencing problem. *Operations Research, 7*, 621–624.

Burkard, R. E., Fortuna, T., & Hurkens, C. A. J. (2002). Makespan minimization for chemical batch processes using non-uniform time grids. *Computers and Chemical Engineering, 26*, 1321–1332.

Castro, P., Barbosa-Póvoa, A.P.F.D., & Matos, H. (2001). An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. *Industrial and Engineering Chemistry Research, 40*, 2059–2068.

Castro, P., Matos, H., & Barbosa-Póvoa, A.P.F.D. (2002). Dynamic modeling and scheduling of an industrial batch system. *Computers and Chemical Engineering, 26*, 671–686.

Cerdá, J., Henning, G. P., & Grossmann, I. E. (1997). A mixed-integer linear programming model for short-term scheduling of single-stage multiproduct batch plants with parallel lines. *Industrial and Engineering Chemistry Research, 36*, 1695–1707.

Cott, B. J., & Macchietto, S. (1989). Minimizing the effects of batch process variability using online schedule modification. *Computers and Chemical Engineering, 13*, 105–113.

Crooks, C. A., Kuriyan, K., & Macchietto, S. (1992). Integration of batch plant design, automation, and operation software tools. *Computers and Chemical Engineering, 16*, S289–S296.

Crooks, C. A., & Macchietto, S. (1992). A combined MILP and logic-based approach to the synthesis of operating procedures for batch plants. *Chemical Engineering Communications, 114*, 117–144.

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

Duran, M. A., & Grossmann, I. E. (1986). An outer approximation algorithm for a class of mixed integer nonlinear programs. *Mathematical Programming, 36*, 307–339.

Elkamel, A., Zentner, M., Pekny, J. F., & Reklaitis, G. V. (1997). A decomposition heuristic for scheduling the general batch chemical plant. *Engineering Optimization, 28*, 299–330.

Floudas, C. A. (1995). *Nonlinear and mixed-integer optimization.* Oxford University Press.

Garey, M. R., & Johnson, D. R. (1979). *Computers and intractability: A guide to the theory of NP-completeness.* New York: W.H. Freeman.

Georgiadis, M. C., Papageorgiou, L. G., & Macchietto, S. (2000). Optimal cleaning policies in heat exchanger networks under rapid fouling. *Industrial and Engineering Chemistry Research, 39*, 441–454.

Glismann, K., & Gruhn, G. (2001). Short-term scheduling and recipe optimization of blending processes. *Computers and Chemical Engineering, 25*, 627–634.

Glover, E. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science, 22*, 455–460.

Grossmann, I. E., & Sargent, R. W. H. (1979). Optimal design of multipurpose chemical plants. *Industrial and Engineering Chemistry Process Design Device, 18*, 343–348.

Hamada, T., & Glazebrook, K. D. (1993). A bayesian sequential single machine scheduling problem to minimize the expected weighted sum of flowtimes of jobs with exponential processing times. *Operations Research, 41*, 924–934.

Harding, S. T., & Floudas, C. A. (1997). Global optimization in multiproduct and multipurpose batch design under uncertainty. *Industrial and Engineering Chemistry Research, 36*, 1644–1664.

Harjunkoski, I., & Grossmann, I. E. (2001). A decomposition approach for the scheduling of a steel plant production. *Computers and Chemical Engineering, 25*, 1647–1660.

Honkomp, S. J., Lombardo, S., Rosen, O., & Pekny, J. F. (2000). The curse of reality—why process scheduling optimization problems are difficult in practice. *Computers and Chemical Engineering, 24*, 323–328.

Honkomp, S. J., Mockus, L., & Reklaitis, G. V. (1999). A framework for schedule evaluation with processing uncertainty. *Computers and Chemical Engineering, 23*, 595–609.

Huercio, A., Espuna, A., & Puigjaner, L. (1995). Incorporating on-line scheduling strategies in integrated batch production control. *Computers and Chemical Engineering, 19*, S609–S615.

Hui, C., & Gupta, A. (2001). A bi-index continuous-time mixed-integer linear programming model for single-stage batch scheduling with parallel units. *Industrial and Engineering Chemistry Research, 40*, 5960–5967.

Hui, C., Gupta, A., & van der Meulen, H. A. J. (2000). A novel MILP formulation for short-term scheduling of multi-stage multi-product batch plants with sequence-dependent constraints. *Computers and Chemical Engineering, 24*, 2705–2717.

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

Ierapetritou, M. G., & Floudas, C. A. (1998b). Effective continuous-time formulation for short-term scheduling. Part 2. Continuous and semi-continuous processes. *Industrial and Engineering Chemistry Research, 37*, 4360–4374.

Ierapetritou, M. G., & Floudas, C. A. (2001). Comments on "an improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants". *Industrial and Engineering Chemistry Research, 40*, 5040–5041.

Ierapetritou, M. G., Hene, T. S., & Floudas, C. A. (1999). Effective continuous-time formulation for short-term scheduling. Part 3. Multiple intermediate due dates. *Industrial and Engineering Chemistry Research, 38*, 3446–3461.

Ierapetritou, M. G., & Pistikopoulos, E. N. (1996). Batch plant design and operations under uncertainty. *Industrial and Engineering Chemistry Research, 35*, 772–787.

Ierapetritou, M. G., & Pistikopoulos, E. N. (1996b). Global optimization for stochastic planning, scheduling and design problems. In I. E. Grossmann (Ed.), *Global optimization in engineering design* (pp. 231–287). Kluwer Academic Publishers.

Jain, V., & Grossmann, I. E. (1999). Resource-constrained scheduling of tests in new product development. *Industrial and Engineering Chemistry Research, 38*, 3013–3026.

Kamburowski, J. (1999). Stochastically minimizing the makespan in two-machine flow shops without blocking. *European Journal of Operations Research, 112*, 304–309.

Kamburowski, J. (2000). On three-machine flow shops with random job processing times. *European Journal of Operations Research, 125*, 440–449.

Kanakamedala, K. B., Reklaitis, G. V., & Venkatasubramanian, V. (1994). Reactive schedule modification in multipurpose batch chemical plants. *Industrial and Engineering Chemistry Research, 33*, 77–90.

Karimi, L. A., & McDonald, C. M. (1997). Planning and scheduling of parallel semi-continuous processes. Part 2. Short-term scheduling. *Industrial and Engineering Chemistry Research, 36*, 2701–2714.

Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operations. Part 1. MELP formulation. *Computers and Chemical Engineering, 17*, 211–227.

Ku, H., & Karimi, L. A. (1988). Scheduling in serial multiproduct batch processes with finite interstage storage: A mixed integer linear programming formulation. *Industrial and Engineering Chemistry Research, 27*, 1840–1848.

Lamba, N., & Karimi, L. A. (2002a). Scheduling parallel production lines with resource constraints. Part 1. Model formulation. *Industrial and Engineering Chemistry Research, 41*, 779–789.

Lamba, N., & Karimi, I. A. (2002b). Scheduling parallel production lines with resource constraints. Part 2. Decomposition algorithm. *Industrial and Engineering Chemistry Research, 41*, 790–800.

Lee, K., Heo, S., Lee, H., & Lee, I. (2002). Scheduling of single-stage and continuous processes on parallel lines with intermediate due dates. *Industrial and Engineering Chemistry Research, 41*, 58–66.

Lee, K., Park, H. I., & Lee, I. (2001). A novel nonuniform discrete time formulation for short-term scheduling of batch and continuous processes. *Industrial and Engineering Chemistry Research, 40*, 4902–4911.

Lin, X., & Floudas, C. A. (2001). Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. *Computers and Chemical Engineering, 25*, 665–674.

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

Lin, X., Chajakis, E. D., & Floudas, C. A. (2003). Scheduling of tanker lightering via a novel continuous-time optimization framework. *Industrial and Engineering Chemistry Research, 42*, 4441–4451.

Lin, X., Janak, S. L., & Floudas, C. A. (2004). A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty. *Computers & Chemical Engineering, 28*, 1069–1085.

Majozi, T., & Zhu, X. X. (2001). A novel continuous-time MILP formulation for multipurpose batch plants. Part 1. Short-term scheduling. *Industrial and Engineering Chemistry Research, 40*, 5935–5949.

Manne, A. S. (1960). On the job-shop scheduling problem. *Operations Research, 8,* 219–223.

Méndez, C.A., & Cerdá, J. (2000a). Optimal scheduling of a resource-constrained multiproduct batch plant supplying intermediates to nearby end-product facilities. *Computers and Chemical Engineering, 2A,* 369–376.

Méndez, C.A., Henning, G. P., & Cerdá, J. (2000b). Optimal scheduling of batch plants satisfying multiple product orders with different due-dates. *Computers and Chemical Engineering, 24,* 2223–2245.

Méndez, C.A., Henning, G. P., & Cerdá, J. (2001). An MILP continuous-time approach to short-term scheduling of resource-constrained multi-stage flowshop batch facilities. *Computers and Chemical Engineering, 25*, 701–711.

Mockus, J., Eddy, W., Mockus, A., Mockus, L., & Reklaitis, R. V. (1997). *Bayesian Heuristic approach to discrete and global optimization: Algorithms, visualization, software, and applications*. Kluwer Academic Publishers.

Mockus, L., & Reklaitis, G. V. (1997). Mathematical programming formulation for scheduling of batch operations based on nonuniform time discretization. *Computers and Chemical Engineering, 21*, 1147–1156.

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

Mockus, L., & Reklaitis, G. V. (1999b). Continuous time representation approach to batch and continuous process scheduling. Part 2. Computational issues. *Industrial and Engineering Chemistry Research, 38*, 204–210.

Moon, S., & Hrymak, A. N. (1999). Mixed-integer linear programming model for short-term scheduling of a special class of multipurpose batch plants. *Industrial and Engineering Chemistry Research, 38*, 2144–2150.

Moon, S., Park, S., & Lee, W. K. (1996). New MILP models for scheduling of multiproduct batch plants under zero-wait policy. *Industrial and Engineering Chemistry Research, 35*, 3458–3469.

Orçun, S., Altinel, I. K., & Hortaçsu, Ö. (1996). Scheduling of batch processes with operational uncertainties. *Computers and Chemical Engineering, 20,* S1191–S1196.

Orçun, S., Altinel, I. K., & Hortacsu, Ö. (2001). General continuous time models for production planning and scheduling of batch processing plants: Mixed integer linear program formulations and computational issues. *Computers and Chemical Engineering, 25*, 371–389.

Pantelides, C. C. (1993). Unified frameworks for optimal process planning and scheduling. In D. W. T. Rip-pin, J. C. Hale, & J. Davis (Eds.), *Proceedings of the second international conference on foundations of computer-aided process operations* (pp. 253–274). Colorado: Crested Butte.

Papageorgaki, S., & Reklaitis, G. V. (1990a). Optimal design of multipurpose batch plants. Part 1. Problem formulation. *Industrial and Engineering Chemistry Research, 29*, 2054–2062.

Papageorgaki, S., & Reklaitis, G. V. (1990b). Optimal design of multipurpose batch plants. Part 2. A decomposition solution strategy. *Industrial and Engineering Chemistry Research, 29*, 2062–2073.

Papageorgaki, S., & Reklaitis, G. V. (1993). Retrofitting a general multipurpose batch chemical plant. *Industrial and Engineering Chemistry Research, 32,* 345–362.

Pekny, J. F., Reklaiti, G. V. (1998). Towards the convergence of theory and practice: A technology guide for scheduling/planning methodology. In J. F. Pekny, & G. E. Blau (Eds.), *Proceedings of the third international conference on foundations of computer-aided process operations* (pp. 91–111). Utah: Snowbird.

Pekny, J. F., Zentner, M. G. (1993). Learning to solve process scheduling problems: The role of rigorous knowledge acquisition frameworks. In D. W. T. Rippin, J. C. Hale, & J. Davis (Eds.), *Proceedings of the second international conference on foundations of computer-aided process operations* (pp. 275–309). Colorado: Crested Butte.

Petkov, S. B., & Maranas, C. D. (1997). Multiperiod planning and scheduling of multiproduct batch plants under demand uncertainty. *Industrial and Engineering Chemistry Research, 36*, 4864–4881.

Pinto, J. M., & Grossmann, I. E. (1994). Optimal cyclic scheduling of multistage continuous multiproduct plants. *Computers and Chemical Engineering, 18*, 797–816.

Pinto, J. M., & Grossmann, I. E. (1995). A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Industrial and Engineering Chemistry Research, 34*, 3037–3051.

Pinto, J. M., & Grossmann, I. E. (1996). An alternate MILP model for short-term scheduling of batch plants with preordering constraints. *Industrial and Engineering Chemistry Research, 35*, 338–342.

Pinto, J. M., & Grossmann, I. E. (1998). Assignment and sequencing models for the scheduling of process systems. *Annals of Operations Research, 81*, 433–466.

Pinto, J. M., Joly, M., & Moro, L. F. L. (2000). Planning and scheduling models for refinery operations. *Computers and Chemical Engineering, 24*, 2259–2276.

Pinto, J. M., Ttirkay, A., Bolio, B., & Grossmann, I. E. (1998). STBS: A continuous-time MILP optimization for short-term scheduling of batch plants. *Computers and Chemical Engineering, 22*, 1297–1308.

Pritsker, A. A. B., Watters, L. J., & Wolfe, P. M. (1969). Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science, 16*, 93–108.

Realff, M. J., Shah, N., & Pantelides, C. C. (1996). Simultaneous design, layout and scheduling of pipeless batch plants. *Computers and Chemical Engineering, 20*, 869–883.

Reklaitis, G. V. (1992). Overview of scheduling and planning of batch process operations. *NATO Advanced Study Institute—Batch process systems engineering*. Turkey: Antalya.

Rippin, D. W. T. (1993). Batch process systems engineering: A retrospective and prospective review. *Computers and Chemical Engineering, 17*, S1–S13.

Rodrigues, M. T. M., Gimeno, L., Passos, C. A. S., & Campos, M. D. (1996). Reactive scheduling approach for multipurpose chemical batch plants. *Computers and Chemical Engineering, 20*, S1215–S1220.

Roslöf, J., Harjunkoski, I., Björkqvist, J., Karlsson, S., & Westerlund, T. (2001). An MILP-based reordering algorithm for complex industrial scheduling and rescheduling. *Computers and Chemical Engineering, 25*, 821–828.

Sahinidis, N. V., & Grossmann, I. E. (1991a). MINLP model for cyclic multiproduct scheduling on continuous parallel lines. *Computers and Chemical Engineering, 15*, 85–103.

Sahinidis, N. V., & Grossmann, I. E. (1991b). Reformulation of multiperiod MILP models for planning and scheduling of chemical processes. *Computers and Chemical Engineering, 15*, 255–272.

Sanmarti, E., Espuna, A., & Puigjaner, L. (1997). Batch production and preventive maintenance scheduling under equipment failure uncertainty. *Computers and Chemical Engineering, 21*, 1157–1168.

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 and Chemical Engineering, 20*, 1263–1268.

Schilling, G. (1997). *Optimal scheduling of multipurpose plants.* Ph.D. thesis, University of London.

Schilling, G., & Pantelides, C. C. (1996). A simple continuous-time process scheduling formulation and a novel solution algorithm. *Computers and Chemical Engineering, 20,* S1221–S1226.

Schilling, G., & Pantelides, C. C. (1996b). A hybrid branch and bound algorithm for continuous-time process scheduling formulations. *AIChE 1996 Annual Meeting*, Chicago, IL, November 1996, Paper 17 Id.

Schilling, G., & Pantelides, C. C. (1999). Optimal periodic scheduling of multipurpose plants. *Computers and Chemical Engineering, 23*, 635–655.

Schweiger, C. A., & Floudas, C. A. (1997). *MINOPT: A software package for mixed-integer nonlinear optimization, user's guide.* New Jersey: Computer-Aided Systems Laboratory, Department of Chemical Engineering, Princeton University .

Shah, N. (1998). Single- and multisite planning and scheduling: Current status and future challenges. In J. F. Pekny, & G. E. Blau (Eds.), *Proceedings of the third international conference on foundations of computer-aided process operations* (pp. 75–90). Utah: Snowbird.

Shah, N., & Pantelides, C. C. (1992). Design of multipurpose batch plants with uncertain production requirements. *Industrial and Engineering Chemistry Research, 31*, 1325–1337.

Shah, N., Pantelides, C. C., & Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operations. Part II. Computational issues. *Computers and Chemical Engineering, 17*, 229–244.

Sparrow, R. E., Forder, G. J., & Rippin, D. W. T. (1975). The choice of equipment sizes for multi-product batch plants. Heuristics vs. branch and bound. *Industrial and Engineering Chemistry Process Design Device, 14*, 197–203.

Subrahmanyam, S., Pekny, J. F., & Reklaitis, G. V. (1994). Design of batch chemical plants under market uncertainty. *Industrial and Engineering Chemistry Research, 33*, 2688–2701.

Suhami, I., & Mah, R. S. H. (1982). Optimal design of multipurpose batch plants. *Industrial and Engineering Chemistry Process Design Device, 21*, 94–100.

Tsirukis, A. G., Papageorgaki, S., & Reklaitis, G. V. (1993). Scheduling of multipurpose batch chemical plants with resource constraints. *Industrial and Engineering Chemistry Research, 32*, 3037–3050.

Vaselenak, J. A., Grossmann, I. E., & Westerberg, A. W. (1987). An embedding formulation for the optimal scheduling and design of multipurpose batch plants. *Industrial and Engineering Chemistry Research, 26*, 139–148.

Vin, J. P., & Ierapetritou, M. G. (2000). A new approach for efficient rescheduling of multiproduct batch plants. *Industrial and Engineering Chemistry Research, 39*, 4228–4238.

Vin, J. P., & Ierapetritou, M. G. (2001). Robust short-term scheduling of multiproduct batch plants under demand uncertainty. *Industrial and Engineering Chemistry Research, 40*, 4543–4554.

Voudouris, V. T., & Grossmann, I. E. (1993). Optimal synthesis of multiproduct batch plants with cyclic scheduling and inventory considerations. *Industrial and Engineering Chemistry Research, 20*, 1962–1980.

Voudouris, V. T., & Grossmann, I. E. (1996). MILP model for scheduling and design of a special class of multipurpose batch plants. *Computers and Chemical Engineering, 20*, 1335–1360.

Wang, S., & Guignard, M. (2002). Redefining event variables for efficient modeling of continuous-time batch processing. *Annals of Operations Research, 116*, 113–126.

Wellons, M. C., & Reklaitis, G. V. (1991a). Scheduling of multipurpose batch chemical plants. Part 1. Formation of single-product campaigns. *Industrial and Engineering Chemistry Research, 30*, 671–688.

Wellons, M. C., & Reklaitis, G. V. (1991b). Scheduling of multipurpose batch chemical plants. Part 2. Multiple-product campaign formation and production planning. *Industrial and Engineering Chemistry Research, 30*, 688–705.

Wilkinson, S. J., Cortier, A., Shah, N., & Pantelides, C. C. (1996). Integrated production and distribution scheduling on a European-wide basis. *Computers and Chemical Engineering, 20*, S1275–S1280.

Xia, Q., & Macchietto, S. (1997). Design and synthesis of batch plants—MINLP solution based on a stochastic method. *Computers and Chemical Engineering, 21*, S697–S702.

Yee, K. L., & Shah, N. (1998). Improving the efficiency of discrete time scheduling formulation. *Computers and Chemical Engineering, 22*, S403–S410.

Yi, G., Suh, K., Lee, B., & Lee, E. S. (2000). Optimal operation of quality controlled product storage. *Computers and Chemical Engineering, 2A*, 475–480.

Zentner, M. G., Elkamel, A., Pekny, J. F., & Reklaitis, G. V. (1998). A language for describing process scheduling problems. *Computers and Chemical Engineering, 22*, 125–145.

Zentner, M. G., Pekny, J. F., Reklaitis, G. V., & Gupta, J. N. D. (1994). Practical considerations in using model-based optimization for the scheduling and planning of batch/semicontinuous processes. *Journal of Process Control, 4*, 259–280.

Zhang, X. (1995). *Algorithms for optimal scheduling using nonlinear models.* Ph.D. thesis, University of London.

Zhang, X., & Sargent, R. W. H. (1996). The optimal operation of mixed production facilities. Part A. General formulation and some solution approaches for the solution. *Computers and Chemical Engineering, 20*, 897–904.

Zhang, X., & Sargent, R. W. H. (1998). The optimal operation of mixed production facilities—extensions and improvements. *Computers and Chemical Engineering, 22*, 1287–1295.