Continuous Time Representation Approach to Batch and Continuous Process Scheduling. 1. MINLP Formulation

L. Mockus and G. V. Reklaitis*

School of Chemical Engineering, Purdue University, West Lafayette, Indiana 47907

The problem of decision timing in the context of batch and continuous process scheduling is addressed in this paper. Representation of time in any scheduling model affects the number of integer variables and the convexity of the model. The usual procedure in batch and continuous process scheduling is to divide the scheduling interval into equal size intervals so as to achieve the required accuracy. This construction generates a formulation with a potentially very large number of binary variables. In this paper, the time events arising in the schedule are modeled directly, and thus the use of binary variables over periods during which no real changes in system state occur is avoided. Furthermore, such a representation of time offers the potential for direct integration of operations scheduling and control which is difficult to achieve in the usual equal size interval approach. The problem is formulated as a mixed integer nonlinear program (MINLP). Issues pertaining to the efficient solution of this problem are discussed in Part 2 of this series.

1. Introduction

A wide range of products in the chemical processing industry are produced using the batch mode of production. This mode of production has long been the accepted procedure for the manufacture of many types of chemicals (specialty chemicals, pharmaceuticals, polymers, biochemicals, and foods), particularly those which are produced in small quantities and for which the production processes or the demand pattern are likely to change. The most important feature of batch processes is their flexibility in processing multiple products by accommodating the diverse operating conditions associated with each product. Therefore, in spite of the traditional drive toward continuous production, the batch mode continues to be the only alternative for a number of sectors of the processing industry. As a result there has been increasing interest in the development of procedures for scheduling batch process operations.

One key consideration in any scheduling algorithm is the representation of time. We consider short term scheduling problems where campaign type, cyclic scheduling techniques are not applicable. The common approach for such problems is to use the uniform time discretization model (UDM). The scheduling horizon is divided into equal size intervals, and any event such as the start or end of a task is allowed to occur only on interval boundaries. There is a binary variable associated with each interval which indicates whether or not that task is started at the beginning of that interval. Thus time is considered as a discrete variable which can attain the values of the beginning of each interval. The main difficulty with this representation is that in order to accurately represent a process we may need to create a model with a very large number of binary variables. To decrease the number of variables, rounding of event times and durations is commonly used. The drawback of rounding is that it is difficult to use such a schedule for process control without *ad hoc* adjustments because the process control logic requires precise execution times. Furthermore, rounding up can produce infeasible or loose schedules while rounding down can produce infeasible schedule. Another inherent UDM difficulty arises in representing continuous processes because of the discrete time representation. A continuous process may start and end somewhere within an equal size UDM interval, not on the interval boundaries. These two limitations are removed by a continuous time representation.

The operations research literature provides a number of mathematical programming formulations of the scheduling problem based on continuous time representation; see Greenberg (1968), for example. An overview of application of this approach to batch process scheduling is presented in Mockus and Reklaitis (1997). In this work the nonuniform time discretization model (NUDM) is proposed to model batch processes. In contrast to the UDM, the binary variables are created only for the real events such as the start or end of a task but not for every artificially created equal size interval. Another key technical difference is that time is considered as a continuous variable. The model is formulated as an MINLP, but exact linearization is used to eliminate some of the nonlinearities. The result is a large scale mixed integer bilinear program which is solved as a series of mixed integer linear programs (MILP). The size of the reformulated problem is the main difficulty with the approach. It was shown, however, that in cases when the uniform discretization interval is small compared to the scheduling horizon length, NUDM performs significantly better than UDM.

A similar approach to that described above was used in Schilling et al. (1994) for multipurpose continuous plant modeling. The time horizon is divided into a number of slots of variable duration. Limited availability of materials, sequence dependent changeovers, and temporary unavailability of processing equipment are taken into account. The problem is formulated as an MINLP, and exact linearization is employed, which

* Author to whom all correspondence should be addressed. E-mail: reklaiti@ecn.purdue.edu. Fax: (317) 494-0805.

10.1021/ie970311r CCC: $18.00
Published on Web 12/04/1998

results in a large scale MILP. Thus the same difficulty, namely a large scale MILP, remains.

The key difference of the present work from Mockus and Reklaitis (1997) and Schilling et al. (1994) is that the NUDM formulation is extended to handle both batch and continuous units, thus, enabling both process types to be modeled within one framework. Furthermore, the proposed numerical algorithm does not use exact linearization, thus avoiding the need for the solution of a series of a large scale MILP's. In some cases not all customer demands can be satisfied as specified and thus have to be satisfied at a later time (soft due dates). The NUDM is also extended to handle this situation. Sequence dependent changeovers are treated differently to be consistent with material balance and allocation handling. The mathematical form of the formulation is radically changed by using a few nonrestrictive assumptions. For modeling purposes we continue to employ the *state task network* formalism (STN) Kondili et al. (1993).

In section 2, we present a rigorous formulation which characterizes mathematically the states and tasks in terms of both integer and continuous variables. The various constraints and objectives under which batch and continuous plants operated are expressed in nonlinear form with respect to these variables.

The formulation presented here results in a mixed integer nonlinear programming (MINLP) problem, in contrast to the MILP obtainable by the UDM approach. However, as will be shown in section 3 with the help of two modest but representative examples, while the size of the MILP produced by the UDM route for a particular problem structure is quite sensitive to the specific problem data, the NUDM formulation is invariant to specific problem data values. Of course, the specific point at which the dimensionality burden will overwhelm the advantage of linearity will depend upon the specific solution algorithms used for the two formulations. In part 2 of this series we will present an NUDM solution algorithm which will demonstrate that tradeoff via computational trials.

2. Mathematical Formulation of the Scheduling Problem

In order to determine optimal schedules of batch and continuous operations, we adopt a particularly simple model for each task. We assume that a task receives material from its feed states in fixed, *a priori* known proportions of its batch size (or processing rate in the continuous case) and that it produces material in its output states also in fixed, known proportions. Furthermore, the processing times for each batch task of each product are also assumed to be independent of the batch size and to be known.

A number of parameters are associated with the tasks and the states defining the STN and with the available equipment items. More specifically, they are as follows.

Task i is defined by ρ_{ijs}, the proportion of input of task i processed in unit j from state s; $\overline{\rho_{ijs}}$, the proportion of output of task i processed in unit j to state s; τ_{ij}, processing time of task i in unit j; $\tau_{ij}^{\min}$, minimum processing time of task i in continuous unit j; J_i^B, set of units capable of performing batch task i; and J_i^C, set of units capable of performing continuous task i. Parameters J_i^B and J_i^C relate the process equipment units to the STN.

State s is defined by T_s, set of tasks receiving material from state s; $\overline{T_s}$, set of tasks producing material of state s; and $S_s^{\max}$, maximum storage capacity dedicated to state s.

Unit j may be capable of performing one or more tasks. It is characterized by I_j, set of tasks which can be performed by unit j; $B_{ij}^{\max}$, maximum capacity of unit j when used for performing task i; $B_{ij}^{\min}$, minimum capacity of unit j when used for performing task i; $r_{ij}^{\max}$, maximum processing rate of continuous unit j when used for performing task i; and $r_{ij}^{\min}$, minimum processing rate of continuous unit j when used for performing task i.

The maximum and minimum unit capacities already take into account any differences in densities between different materials. The maximum capacity obviously reflects the useful size of the vessel. Although the minimum capacity is zero for many common operations, a nonzero value could, for example, represent the minimum volume of liquid required to cover the heating coil in a batch vessel.

The scheduling problem for a batch and continuous processing system can be stated as follows.

Given: the STN of a batch or continuous process and all the information associated with it; and a time horizon of interest.

Determine: the timing of operations for each unit (i.e., which task, if any, the unit performs at any time during the time horizon); the flow of material through the network.

The goal is to *optimize* a given objective criterion involving operating costs, sales revenue, and inventory costs.

For purposes of the current paper, a number of assumptions are made.

(1) Each task occurs at most once in a given equipment unit. This is not an unduly restrictive assumption in the case of batch tasks since one can readily estimate the number of times a task can be executed and thus create a corresponding number of identical tasks. It could be restrictive in some cases of continuous tasks (*i.e.*, when minimum processing time is zero).

(2) The output from a task to all its output states occurs at the same time. To model separation processes where this is not the case, we create a chain of tasks with processing times equal to the corresponding difference in output times. States which connect these tasks are then required to be of the zero wait type.

(3) No preemptive operation is allowed (i.e. no task can be interrupted, once started).

(4) The material is transfered instantaneously from states to tasks and from tasks to states.

(5) All data are deterministic and fixed over the time horizon of interest.

With these assumptions we are now in position to proceed to the mathematical formulation of the scheduling problem described above.

We base our formulation on a continuous time representation in which the time horizon of interest is divided into number of intervals of unequal duration. Events of any type—such as the start or end of processing individual batches of individual tasks, changes in availability of processing equipment and other resources etc.—are only allowed at the interval boundaries. We number the intervals from 1 to H, where H is the total number of events (see Figure 1).

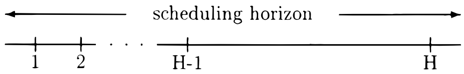

Figure 1. Time representation.

On the basis of the above time representation, the following variables are introduced to characterize the tasks in the STN: t_o, time corresponding to interval o; $W_{ijo}^S = 1$ if unit j starts processing task i at the beginning of interval o but 0 otherwise; $W^F{}_{ijo} = 1$ if unit j ends processing task i at the beginning of interval o but 0 otherwise; $W_{ijo} = 1$ if unit j is processing task i during interval o but 0 otherwise; B_{ij}, batch size of task i which is processed in unit j. Q_{ijo}, the amount of material produced by task i in continuous unit j at the end of interval o.

In addition to characterizing the tasks, it is also necessary to characterize the states and equipment units, through the introduction of the following variables: S_{so}, amount of material stored in state s at the beginning of interval o; N_{jo}, number of available units j at the beginning of interval o.

Having identified the variables of the system, we need to express the system limitations into explicit mathematical constraints and the system performance criterion into an explicit objective function.

2.1. Constraints. There are three types of fundamental constraints which occur in all batch and continuous scheduling problems: (1) material balances; (2) capacity constraints—limitations on the capacities of units, storage vessels, and processing rates; (3) allocation constraints—the resolution of conflicts when equipment items are allocated to tasks.

2.1.1. Material Balances. Material balances are expressed mathematically as follows:

$$S_{so} = S_{s,o-1} - \sum_{i \in T_s} \sum_{j \in J_i} \rho_{ijs} (B_{ij} W_{ijo}^S + Q_{ijo}) + \sum_{i \in \overline{T_s}} \sum_{j \in J_i} \overline{\rho_{ijs}} (B_{ij} W_{ijo}^F + Q_{ijo}), \ \forall s, o \quad (1)$$

$$\tau_{ij} = \sum_o t_o (W^F{}_{ijo} - W_{ijo}^S), \ \forall i, j \in J_i^B \quad (2)$$

$$\tau_{ij}^{\min} \le \sum_o t_o (W_{ijo}^F - W_{ijo}^S), \ \forall i, j \in J_i^C \quad (3)$$

Constraint (1) simply states that the net increase ($S_{so} - S_{s,o-1}$) in the amount of material stored in a state s at the beginning of interval o is given by the difference of the amount produced in this state and that used. The initial amount S_{s0} of material in each state s is assumed to be known, thus allowing the precise initial condition of all material inventories (including intermediates and final products) to be specified. Constraints (2) and (3) ensure that the time elapsed between the task start and end is equal to task processing time τ_{ij} for batch tasks; it is greater than the minimum run time $\tau_{ij}^{\min}$ for continuous tasks.

It is often necessary (e.g. due to contractual obligations) to deliver to customers certain agreed quantities D_{sl}, $l = 1, ..., D_s$ of material in product state s at various times t_{sl}^D. Furthermore, it may be necessary (e.g. due to limited availability of local storage capacity) to receive quantities R_{so} of raw materials in feed state s at the beginning of interval o during the schedule rather than having all the required feedstock stored locally at the start of processing.

These complications can readily be incorporated in the mathematical formulation by modifying the material balance constraint (1)

$$S_{so} = S_{s,o-1} - \sum_{D_s}^{l=1} W_{slo}^D D_{sl} + R_{so} - \sum_{i \in \overline{T_s}} \sum_{j \in J_i} \overline{\rho_{ijs}} (B_{ij} W_{ijo}^F + Q_{ijo}) + \sum_{i \in \overline{T_s}} \sum_{j \in J_i} \overline{\rho_{ijs}} (B_{ij} W_{ijo}^F + Q_{ijo}), \ \forall s, o \quad (4)$$

$$\sum_o W_{slo}^D = 1 \text{ and } \sum_o t_o W_{slo}^D = t_{sl}^D, \ \forall s, l = 1, ..., D_s \quad (5)$$

where $W_{slo}^D = 1$ if delivery D_{sl} occurs at the beginning of interval o. The second term in constraint (4) accounts for the delivery of the material from the certain state s (since this material is removed from the system, a minus sign must be inserted before this term). The third term accounts for the feeds (in this case the plus sign is used to represent the fact that the material is an input to the system). The constraint (5) forces the specific delivery D_{sl} to occur only once and the time of the delivery t_o to be equal to t_{sl}^D. The batch and continuous tasks are treated differently because material input and output for batch processes occurs only during the start and end of a particular task while for continuous processes the transfer of the material occurs continuously.

2.1.2. Capacity Constraints. The amount of material that starts undergoing task i in unit j is bounded by the maximum and minimum capacities of that unit:

$$B_{ij}^{\min} \sum_o W_{ijo}^S \le B_{ij} \le B_{ij}^{\max} \sum_o W_{ijo}^S, \ \forall i, j \in J_i^B \quad (6)$$

The same is true for processing rates:

$$r_{ij}^{\min} W_{ijo}(t_{o+1} - t_o) \le Q_{ijo} \le r_{ij}^{\max} W_{ijo}(t_{o+1} - t_o), \ \forall i, j \in J^C{}_i, o \quad (7)$$

The amount of material stored in a state s must not at any time exceed the maximum storage capacity for this state:

$$0 \le S_{so} \le S_s^{\max}, \ \forall s, o \quad (8)$$

2.1.3. Allocation Constraints. At any time, an idle item of equipment can only start at most one task. Of course, if the item does start performing a given task, then it cannot start any other task until the current one is finished; i.e., the operation is nonpreemptive. Similarly, the task once started must be completed in the same equipment unit. All these requirements can be expressed in terms of the following constraints

$$N_{jo} = N_{j,o-1} - \sum_{i \in I_j} W_{ijo}^S + \sum_{i \in I_j} W_{ijo}^F, \ \forall j, o \quad (9)$$

$$0 \le N_{jo} \le 1, \ \forall j, o \quad (10)$$

where $N_{j0} = 1$. Constraint (9) is similar to (1). This constraint is a resource balance on the number of available units. It states that the net increase in the available units ($N_{jo} - N_{j,o-1}$) is the difference between the number of units which finished (the last term) and began (the second term) processing. One may imagine having a pool of available units. When some unit begins processing it is removed from the pool; when it finishes processing, it is returned back to the pool. Constraint (10) requires the number of available units to be between zero and one at any time, thus enforcing that an idle item of equipment can only start at most one task and that the operation is nonpreemtive. It should be noted that this form of resource balance is similar to the form introduced by Pantelides (1994).

Furthermore, during the time horizon of interest, certain items of equipment may become unavailable due to maintenance or breakdowns. Assume that unit j becomes unavailable (or "dies") at times t^D_{jl}, $l = 1, ..., D_j$ and becomes available (or is "born") at times t^B_{jl}. This feature can be taken into account by modifying the allocation constraint (9)

$$N_{jo} = N_{j,o-1} - \sum_{i \in I_j} W^S_{ijo} + \sum_{i \in I_j} W^F_{ijo} - \sum_{l=1}^{D_j} W^D_{jlo} + \sum_{l=1}^{D_j} W^B_{jlo}, \ \forall j, o \quad (11)$$

$$\sum_o W^D_{jlo} = 1 \quad \text{and} \quad \sum_o t_o W^D_{jlo} = t^D_{jl}, \ \forall j, l = 1, ..., D_j \quad (12)$$

$$\sum_o W^B_{jlo} = 1 \quad \text{and} \quad \sum_o t_o W^B_{jlo} = t^B_{jl}, \ \forall j, l = 1, ..., D_j \quad (13)$$

where $W^D_{jlo} = 1$ if unit j "dies" at the beginning of interval o and $W^B_{jlo} = 1$ if it is "born" at the beginning of interval o. The two last terms in constraint (11) account for the "death" of the unit (unit is removed from the pool of the available units) and "birth" of the unit (unit is entered into the pool of available units). Constraint (12) states that the unit j may "die" (or become unavailable for processing due to the breakdown or maintenance) only at specific times t^D_{jl}. Constraint (13) is similar, except that it accounts for the "birth" of the unit j.

2.1.4. Limited Availability of Utilities and Manpower. In addition to processing equipment, the tasks in a recipe may require the use of utilities (e.g., steam, electricity, cooling water etc.), and/or manpower. Furthermore, at any given time, the amount required may be constant or it may depend on the batch size (processing rate).

We assume that the amount of utility u required by task i is given by the combination of a constant and a variable term of the form

$$\alpha_{uij} + \beta_{uij} B_{ij} + \beta_{uij} Q_{ijo}, \ \forall u, i \in I_u, j \in J_i, o \quad (14)$$

where B_{ij} is the relevant batch size, Q_{ijo} is the relevant amount of material produced by a continuous task, and I_u is the set of tasks which require utility u. For manpower and other discrete utilities (such as auxiliary equipment), the constant demand factors α_{uij} are normally integer, while the variable factors β_{uij} are likely to be zero.

The total demand U_{uo} for utility u over interval o is given by

$$U_{uo} = \sum_{i \in I_u} \sum_{j \in J_i} W_{ijo}(\alpha_{uij} + \beta_{uij} B_{ij} + \beta_{uij} Q_{ijo}), \ \forall u, o \quad (15)$$

If the task i is processed on unit j at the time t_o then W_{ijo} is 1 and the utility u consumption is equal to the sum of constant and variable terms. The maximum amount of utility $U^{\max}_u$ at any time cannot be exceeded by the total demand. This leads to the following simple bounds:

$$U_{uo} \le U^{\max}_u, \ \forall u, o \quad (16)$$

2.1.5. Sequence Dependent Cleaning. It is often the case in batch plants that certain equipment units require cleaning or some other preparation between uses. As with ordinary tasks, we assume that cleaning operations are of fixed duration, during which they may place demands on utilities and manpower. In the sequence-dependent cleaning case, the type and duration of cleaning required between two tasks depends on the identity and relative order of these two tasks. For instance, in a dye manufacturing plant, a unit may require extensive cleaning if the processing of a dark dye is to be followed by that of a lighter one. However, little or no cleaning may be required in the reverse situation.

We create a cleaning task $j_{kk'}$ with processing time $\tau_{j_{kk'}j}$ to denote the cleaning of unit j required between tasks k and k'. For instance, in the dye example mentioned above, we could have a "dark" and a "light" tasks, with, say, $\tau_{j_{12}j} = 2h$ and $\tau_{j_{21}j} = 0$. The utility demand factors $\alpha_{uj_{kk'}j}$ and $\beta_{uj_{kk'}j}$ can then be set to values reflecting the actual requirements of the cleaning operation.

To model this situation we introduce the following variables: $N^C_{j_k o}$, number of units j ready to execute task k at the beginning of interval o ("clean" units); $N^D_{j_k o}$, number of units j which finished processing task k at the beginning of interval o ("dirty" units); $W^S_{j_{kk'}jo} = 1$ if cleaning operation between tasks k and k' is started but 0 otherwise; $W^F_{j_{kk'}jo} = 1$ if cleaning operation between tasks k and k' is finished but 0 otherwise.

Then we can express cleaning constraints in the following way:

$$N^C_{j_k o} = N^C_{j_k,o-1} - W^S_{kjo} + \sum_{k':k' \ne k} W^F_{j_{k'k}jo}, \ \forall j, k \in I_j, o \quad (17)$$

$$N^D_{j_k o} = N^D_{j_k,o-1} + W^F_{kjo} - \sum_{k':k' \ne k} W^S_{j_{k'k}jo}, \ \forall j, k \in I_j, o \quad (18)$$

$$N_{jo} = \sum_{k \in I_j} (N^C_{j_k o} + N^D_{j_k o}), \ \forall j, o \quad (19)$$

$$\tau_{j_{kk'}j} = \sum_o t_o (W^F_{j_{kk'}jo} - W^S_{j_{kk'}jo}), \ \forall j, k \ne k' \in I_j \quad (20)$$

where $N^F{}_{jk0} = 0$. Constraint (17) states that the increase in the number of "clean" units ready to process a task ($N^C_{j_k o} - N^C_{j_k,o-1}$) is given by the difference in the number of the "cleaned" units (the last term) and the number of the units which began processing (the second term). The increase in the number of "dirty" units ($N^D_{j_k o} - N^D_{j_k,o-1}$) is given in a similar manner by the difference in the number of the units which finished processing (the second term in constraint (18) and the units which began to be "cleaned" (the last term). Constraint (19) gives the number of all available units. These three constraints readily replace the allocation constraint (9).

Constraint (20) requires that the duration of the cleaning task be equal to $\tau_{j_{kk'}j}$. The cleaning representation is similar to constructions used by Crooks et al. (1993).

Of course, if a cleaning task begins, it has to end:

$$\sum_o W^S_{j_{kk'}jo} = \sum_o W^F_{j_{kk'}jo} = \min\{\sum_o W^S_{kjo}, \sum_o W^S_{k'jo}\}, \forall j, k \neq k' \in I_j \quad (21)$$

Although constraints (17−20) do not actually place an upper limit on the number of cleaning operations performed, this is normally kept to a minimum by the optimization process itself due to the cost and/or non availability of utilities, and the restrictions on the availability of equipment imposed by cleaning.

In practice, the use of (17−20) can lead to very large numbers of binary variables [$O(t^2)$, where t is the number of tasks]. In many cases, it may be sufficient to ensure that adequate time is left for a unit to be cleaned between uses, without attempting to define the precise timing of the cleaning operations, or to take into account any demands that these operations may pose on utilities. In such cases, it is enough to guarantee that, if the unit starts processing any task k, no task k' can start at least $\tau_{j_{kk'}j}$ after the end of the first task. This can be written as

$$\sum_{m=o+1} t_m W^S_{k'jm} \leq W^F_{kjo}\tau_{j_{kk'}j}, \forall j, k' \neq k, o \quad (22)$$

2.1.6. Miscellaneous Constraints. We need some additional constraints which relate binary variables W_{ijo}, W^S_{ijo}, and W^F_{ijo}. The first of these ensures that $W_{ijo} = 1$ only during processing of task i in unit j

$$W^S_{ijo} \leq W_{ijo}$$
$$W^S_{ijo} \leq 1 - W_{ij,o-1}, \forall i, j \in J_i, o, \quad (23)$$
$$W^S_{ijo} \geq W_{ijo} - W_{ij,o-1}$$

where $W_{ijo} = 0$.

The following constraint

$$\sum_o W^S_{ijo} = \sum_o W^F_{ijo} \leq 1, \forall i, j \in J_i \quad (24)$$

represents the fact that once a task is started, it has to be finished.

2.2. Objective Function. The model is capable of accommodating a variety of either economic or system performance measures. The criterion used in the present study is the maximization of profit. The profit can be expressed as

$$\text{profit} = \text{value of products} - \text{cost of feedstocks} - \text{cost of storage} - \text{cost of utilities} - \text{cost of cleaning} - \text{penalty for missed demands} \quad (25)$$

Each of these terms is quantified as follows:

$$\text{value of products} \equiv \sum_s c_s(\sum_{l=1}^{D_s} D_{sl} + S_{sH}) \quad (26)$$

where c_s is the unit price associated with material in state s. We note that the above expression includes both the value of material left in plant storage at the end of the horizon, and the value of material delivered to customers during the horizon.

If it is undesirable for material in certain (e.g., intermediate) states to be left in storage at the end of the time horizon, this can be avoided either by setting the corresponding c_s to large negative values or by explicitly adding the constraint $S_{sH} = 0$ to the mathematical formulation.

$$\text{cost of feedstocks} \equiv \sum_s c_s(S_{s0} + \sum_o R_{so}) \quad (27)$$

Again, both the value of the initial inventory, and the cost of material received during processing are taken into account. It is worth noting here that the quantities R_{so} may be either fixed *a priori* or allowed to vary, subject to given upper bounds. In the later case, we will determine the optimal schedule of obtaining feedstocks by taking advantage of any demand variations with time and exploiting the available local storage to the best possible degree. The cost of including this feature is minimal since R_{so}s are continuous variables occurring linearly in the constraints and the objective function.

$$\text{cost of storage} \equiv \sum_s c^S_s \sum_o S_{so}(t_{o+1} - t_o) \quad (28)$$

where c^S_s is the running cost of storing a unit amount of material in state s in unit time, H is the scheduling horizon, and $t_{H+1} = H$. Such a cost may, for instance, be incurred if refrigerated or heated storage facilities are required for certain states. The cost of storage is equal to the sum of all cost of storages incurred for storing material during the time interval (t_o, t_{o+1}), which, in turn, is equal to the product of the amount of the material stored and the time interval length. We have to note that this expression is exact only for batch processes since the amount of material stored in state s remains constant for the time interval (t_o, t_{o+1}). However, it is a good approximation for continuous processes when the cost of storage is small compared to the value of products since the amount of the material stored changes continuously within the given time period.

We assume that the cost incurred by processing the task i in unit j is given by the combination of a constant and a variable term of the form

$$\alpha_{ij} + \beta_{ij}B_{ij} + \beta_{ij}\sum_o Q_{ijo}$$

where B_{ij} is the relevant batch size and $\sum_o Q_{ijo}$ is the relevant amount of material produced by task i on unit j. Then

$$\text{cost of utilities} \equiv \sum_j \sum_{i \in I_j}(\alpha_{ij} \sum_o W_{ijo} + \beta_{ij}B_{ij} + \beta_{ij} \sum_o Q_{ijo}) \quad (29)$$

We also assume that the cost incurred by cleaning task $j_{kk'}$ is given by the combination of a constant and a variable term of the form

$$\alpha_{j_{kk'}j} + \beta_{j_{kk'}j}B_{kj} + \beta_{j_{kk'}j} \sum_o Q_{kjo}$$

where B_{kj} is the relevant batch size and $\sum_o Q_{kjo}$ is the

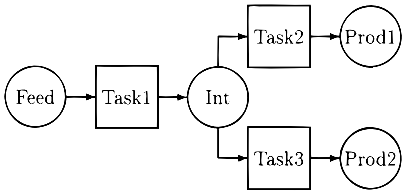

Figure 2. State task network, for example BATCH.

Table 1. Data Used for Example BATCH1

States

states	capacity limits	prices
feed	unlimited	5
int	5000	
prod1	unlimited	10
prod2	unlimited	8

Costs

$\alpha_{ij} = 200$	$\beta_{ij} = 0.6$	$c_s^M = 0.18$

Demands

states	time 4	6	7	10	11	12
prod1	200		300	400	100	
prod2	50	150		200		100

Units, Tasks

units	size	units suitability	processing times
unit1	1500	task1	1 (0.9)
unit2	1000	task2	1 (1.1)
unit3	1000	task3	1 (0.8)

relevant amount of material produced by the task k on unit j. Then

$$\text{cost of cleaning} \equiv \sum_j \sum_{k,k' \in I_j, k \neq k'} \sum_o W^S_{j_{kk'}jo}(\alpha_{j_{kk'}j} + \beta_{j_{kk'}j} B_{kj} + \beta_{j_{kk'}j} r_{kj}) \quad (30)$$

In many instances, demands cannot be satisfied as specified. This situation is accommodated by penalizing for missed demands, where the penalty parameter reflects the priority and importance of the demand, with high penalty values corresponding to demands which must be met. We also assume that each demand has to be satisfied in full. Then

$$\text{penalty for missed demands} \equiv \sum_s \sum_{l=1}^{D_s} D_{sl}(c_s^{DF} + C_s^{DV}(t_{sl}^D - \overline{t_{sl}^D})) \quad (31)$$

where c_s^{DF} = fixed penalty for a missed demand, c_s^{DV} = variable penalty, and $\overline{t_{sl}^D}$ = initial demand time.

3. Illustrative Example

In order to illustrate the relative merits of the proposed formulation vis a vis the conventional UDM formulation, we consider two examples. The first of these is BATCH1 from Sahinidis and Grossmann (1991). The parameters of this problem are shown in Table 1, and the recipe network is shown in Figure 2.

We consider the example in both its original and a modified form in which the uniform time discretization interval is selected to be 0.1, that is, is much smaller than the scheduling horizon (12). This would naturally occur if the processing times were not all unity but instead take on decimal values, e.g., 0.9, 1.1, and 0.8, and one sought to obtain an exact schedule rather than one in which the processing times are rounded to the nearest integer. This is important when, for instance, one seeks to actually transfer the resulting schedule for automated execution. The key model dimensions for BATCH1 under both formulations and data sets are shown in Table 2.

Table 2. UDM and NUDM Comparison

problem	UDM size bin	UDM size cont	NUDM size bin	NUDM size cont
original	33	93	168	99
modified	305	860	168	99

Table 3. UDM and NUDM Comparison for an Example with Sequence Dependent Changeovers

model	changeover time (min)	discretization length (min)	variables bin	variables cont	constraints
UDM	60	60	60	95	289
NUDM			36	45	57
UDM	90	30	109	174	777
NUDM			36	45	57
UDM	75	15	209	334	2432
NUDM			36	45	57
UDM	66	6	509	814	12797
NUDM			36	45	57

Note that for the problem form with unit processing times, the UDM formulation requires only 33 binary variables vs 168 for the NUDM formulation. However, as the time discretization interval is reduced to accommodate the modified processing times, the UDM formulation grows to 305 binary variables, while the NUDM remains unchanged.

As a further illustration we consider the single unit sequencing problem in which given amounts (75 kg each) of three products (white, gray, and black dye) must be produced within a fixed time period (6 h). Each product requires execution of a single task with a processing time of one hour. Moreover, after utilization the unit requires a 1 h clean-out time in preparation for the next task. In this case, the UDM formulation can be constructed using a 1 h time discretization length. In addition, we consider three further cases in which the clean-out time is 90, 75, and 66 min. The UDM formulation will thus require time discretization lengths of 30, 15, and 6 min, respectively. The resulting comparison of formulation dimensionalities is shown in Table 3.

As can be seen, the UDM formulation will grow from 60 binary variables to 509 binary variables, with a corresponding growth in both number of continuous variables and a dramatic growth in constraints, from 297 to almost 13 000. By contrast, the NUDM formulation remains unchanged in number of binary variables, continuous variables, as well as constraints.

From these illustrations, it can be seen that although the NUDM formulation suffers the disadvantage of nonlinearity, the strong dependence of the size of the UDM formulation on the problem parameters indicates that a crossover in solution efficiency will generally be attained as the accuracy in the representation of process times is increased. Of course, the particular point at which NUDM formulation will dominate the UDM formulation in actual computing time will depend on the particular solution algorithms employed. In part 2 of this series, we will present some computational results for a specific solution approach to the NUDM formulation which will make this point explicit.

4. Conclusions

This paper presents a general formulation for planning the operation of batch or continuous plants which simultaneously considers the sequencing and scheduling problems. Furthermore, the approach allows the direct integration of operations scheduling and control by not sacrificing accuracy.

The problem has been formulated mathematically as a MINLP model, the solution of which provides the optimal allocation of equipment to tasks, the associated batch sizes and processing rates, and the optimal utilization of any available storage capacity and scarce utilities and manpower. A general measure of the economic performance of the plant is used as the objective function in this paper.

The formulation presented is aimed at the short-term deterministic scheduling problem. In a real plant, task processing times may fluctuate around some mean value due to impurities in the feed and other operating conditions. Customer demands may be uncertain also. Extension of the formulation to handle these and other uncertainties will be presented in future publications.

The main difficulty of the proposed formulation is the size of the resulting MINLP problem. Even the solution of a small example using a generic solver such as GAMS/DICOPT may require substantial amounts of computation. Furthermore, the computational cost increases rapidly with problem size. For realistic scheduling problems, the MINLP generated may involve several thousands of binary variables, which by far exceeds the solution capabilities of general-purpose methods that are currently available. Therefore, dealing with such large problems requires careful exploitation of the specific features of the problem at hand. These issues are examined in detail in part 2 of this series, see Mockus and Reklaitis (1999).

Literature Cited

Crooks, C. A.; Shah, N.; Pantelides, C. C.; Macchietto, S. Detailed scheduling of general batch and semi-continuous process operations. Technical Report, B93-05; Centre for Process Systems Engineering, Imperial College: London, SW7 2BY, UK, 1993.

Greenberg, H. A branch and bound solution to the general scheduling problem. *Oper. Res.* **1968**, *16*, 13−17.

Kondili, E.; Pantelides, C. C.; Sargent, R. W. H. A general algorithm for short-term scheduling of batch operations. I. MILP formulation. *Comput. Chem. Eng.* **1993**, *17*, 211−227.

Mockus, L.; Reklaitis, G. V. Mathematical programming formulation for scheduling of batch operations based on nonuniform time discretization. *Comput. Chem. Eng.* **1997**, *22*, 1147−1156.

Mockus, L.; Reklaitis, G. V. Continuous time representation approach to batch and continuous process scheduling - II. Computational issues. *Ind. Eng. Chem. Res.* **1999**, *38*, 204−210.

Pantelides, C. C. Unified frameworks for optimal process planning and scheduling. In *Proceedings of the Second International Conference on Foundations of Computer Aided Process Operations*; CACHE Publications: Austin, TX, 1994.

Sahinidis, N. V.; Grossmann, I. E. Reformulation of multiperiod MILP models for planning and scheduling of chemical processes. *Comput. Chem. Eng.* **1991**, *15*, 255−272.

Schilling, G.; Pineau, Y. E.; Pantelides, C. C.; Shah, N. Optimal scheduling of multipurpose continuous plants. Presented at the AIChE Annual Meeting, San Francisco, CA, Nov 1994; Paper 233i.

Received for review May 2, 1997
Revised manuscript received October 16, 1998
Accepted October 19, 1998

IE970311R