INT. J. PROD. RES., 1996, VOL. 34, NO. 2, 525–541

Scheduling multi-purpose batch plants with junction constraints

Y. M. DESSOUKY†, C. A. ROBERTS‡, M. M. DESSOUKY§*
and G. WILSON¶

There is an increasing trend in the chemical process industry to operate flexible batch plants because of their capability to manufacture multiple products simultaneously by sharing the same process resources. In this paper, the scheduling of multi-purpose batch chemical plants with junction (header) constraints is considered. A mixed-integer non-linear model for the scheduling of multi-purpose batch chemical plants is formulated that considers the connection between equipment sets, transfer times, variable batch sizes, alternative process plans, and batch merging. Because of the computational time complexity of the batch scheduling problem, a heuristic scheduling algorithm that minimizes the total tardiness is developed to solve the model.

1. Introduction

There is an increasing trend in the chemical process industry to operate flexible batch plants because of their capability to manufacture multiple products simultaneously by sharing the same process resources. Emphasis on planning and scheduling is greatly increased for batch plants in order to make effective use of this capability (Rippin 1991 and Verhulst 1989). In this paper, the scheduling of batch chemical plants with junction (header) constraints is considered.

Batch chemical plants are classified as either *multi-product* or *mutli-purpose* (Rippin 1983). Multi-product batch chemical plants are analogous to flow-shops in discrete manufacturing where all products follow the same sequence of operations (a single process path). Multi-purpose batch chemical plants are analogous to job-shops with alternative routeing in discrete manufacturing where products may follow different process paths. Complicating the scheduling of multi-purpose plants is that in most cases no buffer is allowed between different equipment sets (machines). A batch cannot be transferred to a machine performing the next operation unless that machine is currently idle. In addition, usually more than one batch cannot be transferred through a header (junction) simultaneously. For this reason, it is important that batch transfer activities be closely coordinated with the start of operation processing activities in order to make efficient use of the resources.

Because of the computational time complexity of the batch scheduling problem, there is a trend to focus on efficient heuristic procedures that yield suboptimal solutions to the problem (Verhulst 1989 and Ku *et al.* 1987). There is a significant amount of literature on efficient solution procedures for the scheduling of multi-product batch

Received October 1994.

† Systems Analysis Department, Miami University, Oxford, OH 45056, USA.
‡ Department of Industrial and Management Systems Engineering, Arizona State University, Tempe, AZ 85287-5906, USA.
§ Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, CA 90089-0193, USA.
¶ Honeywell Industrial Automation and Control, Phoenix, AZ 85023, USA.
* To whom correspondence should be addressed.

plants. To name a few, Ku and Karimi (1988), Birewar and Grossmann (1989), and Tandon *et al.* (1991) report on solution procedures for the makespan objective function. Ku and Karimi (1990, 1991) and Birewar and Grossmann (1990) report on solution procedures that minimize the tardiness penalty.

Kondili *et al.* (1993) point out that there is limited research in the development of heuristic solution procedures for scheduling multi-purpose batch plants, as compared to research in scheduling discrete-parts manufacturing operations. The authors suggest that the limited research is due to the 'additional complexity of the operations involved, and continuous nature of the material being handled'. Suhami and Mah (1982) develop a heuristic solution procedure based on a linear programming model to minimize the total tardiness for the scheduling of production runs of identical batches. Egli and Rippin (1986) develop an enumerative solution procedure for the scheduling of non-identical batches. Because the solution procedure evaluates all possible sequences, the computational requirements can be extensive. Rich and Prokopakis (1986) and Patsidou and Kantor (1991) formulate the problem as a mixed-integer non-linear program. The models assume transfer times are zero or are incorporated in the operation processing time, the number of batches to be scheduled is fixed (a single batch for each product), and batches do not merge, or split.

Musier and Evans (1991) argue that assuming a fixed number of batches is not reasonable because most customer orders are either larger or smaller than the batch size of the plant's process units. More than one batch may be required to satisfy the demand of a particular order, or one batch may satisfy the demand of several orders requiring the same product type. For this reason, the batch sizes are variable and need to be determined by the scheduling system. Musier and Evans develop a heuristic solution procedure that considers variable batch sizes to minimize the total tardiness of the orders, but the procedure is limited to a single-stage system. Kondili *et al.* (1993) develop a mixed-integer linear program for multiple-stages considering variable batch sizes and batch splitting and merging. Their model maximizes profit which is a function of the value of the product, and the cost of raw materials, utilities and material storage. Their model assumes there is a connection between all equipment sets and instantaneous transfer of batches. For some systems, it is unrealistic to assume instantaneous transfer of batches because the same header (junction) may connect more than one machine, leading to competing transfer requests at the same time. In this case the transfer and waiting times for the junctions are important. The modelling of the transfer process is particularly important for merged batches because more than one transfer through the same junction is required.

In this paper, the concept of a junction resource to prevent multiple batches from simultaneously being transferred through a junction is introduced. A mixed-integer non-linear model for the scheduling of multi-purpose batch chemical plants is formulated that considers the connection between equipment sets, transfer times, variable batch sizes, alternative process plans, and batch merging. Because of the computational time complexity of the batch scheduling problem, a heuristic scheduling algorithm that minimizes the total tardiness is developed to solve the model. The objective of the algorithm is to minimize the total tardiness rather than the makespan or flowtime because the total tardiness of orders is an important customer concern (Ku and Karimi 1990). The proposed algorithm determines the size of each batch and the schedule of batches for each resource in the plant to satisfy the demand of all outstanding orders. Each customer order specifies the desired product type, the order quantity, and the due date.

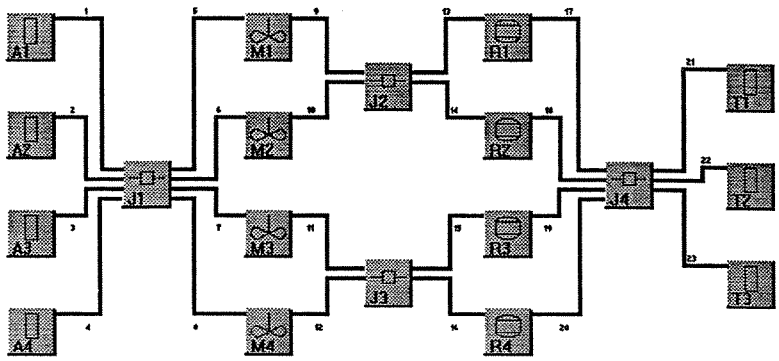

Figure 1. Sample configuration of multi-purpose batch plant.

Equipment name	Type	Capacity (kgs)
Tank-A1	—	infinite
Tank-A2	—	infinite
Tank-A3	—	infinite
Tank-A4	—	infinite
Mixer-M1	stainless-steel	1000
Mixer-M2	stainless-steel	1000
Mixer-M3	glass	2000
Mixer-M4	glass	2000
Reactor-R1	stainless-steel	2000
Reactor-R2	stainless-steel	2000
Reactor-R3	stainless-steel	4000
Reactor-R4	glass	4000
Tank-T1	—	infinite
Tank-T2	—	infinite
Tank-T3	—	infinite

Table 1. Equipment type and characteristics.

2. Batch chemical manufacturing

An actual configuration of a multi-purpose batch chemical process plant is shown in Fig. 1. This configuration is used to manufacture multiple chemical consumer goods using the same equipment resources. The main objective is to blend multiple ingredients to produce an end product. A product is a function of the ingredients, the equipment resources, and the sequence of operations to produce the batch.

The boxes in Fig. 1 represent the resources in the plant and the lines represent the pipes connecting the various resources. There are equipment (machine) and header (junction) resources. The equipment resources are *tanks*, *mixers*, and *reactors*. The tanks store the various chemicals and may contain only one product type at a time. Tanks with an *A* prefix store the raw chemicals (four tanks) and tanks with a *T* prefix store the final products (three tanks). The mixers prepare the premix recipe for use by the reactor and have an *M* prefix (four mixers). The reactors perform the main mixing and reaction operation and have an *R* prefix (four reactors). Some operations require a specific mixer or reactor type while other operations are indifferent to the mixer or reactor type. Possible mixer and reactor types are glass and stainless-steel. Table 1

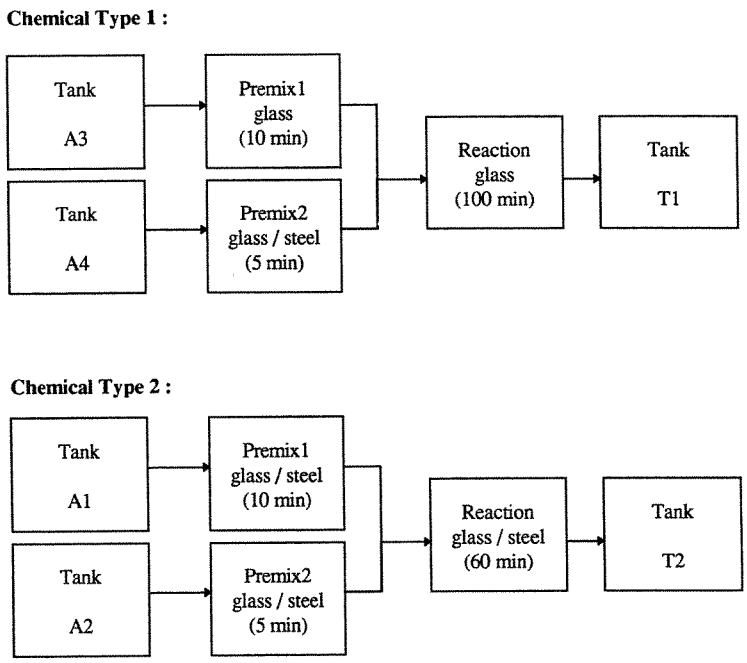

Figure 2. Sample recipes.

summarizes the characteristics of the equipment resources in the plant including the type and capacity. It should be noted that the modelling approach used in this paper is independent of the equipment type in the plant and can be used to model equipment other than tanks, mixers, or reactors such as distillation columns, furnaces, etc.

The junction (headers) resources serve as connectors to transfer chemicals between the equipment resources and start with a *J* prefix (four junctions). The headers are modelled as a resource because it is assumed that only one batch can be transferred through a header at a time. The transfer time is a function of the transfer rate through the header and the size of the batch.

Each chemical product is defined by a *recipe* or set of operations that make up the manufacturing process of a particular product type. Figure 2 shows a sample recipe for two different products with each box representing a specific operation in the recipe. For product type 1, raw materials are derived from two separate tanks (A3 or A4), with each tank holding half of the raw material to make up the final product. The material in tank A3 must be charged (premixed) by glass mixer M3 or M4 for 10 minutes and then transferred to the glass reactor (R4). The material that is in tank A4 can be charged by any mixer (M1, M2, M3, M4) and is processed for 5 minutes, then transferred to the glass reactor (R4). Once the material has been transferred to the reactor (from both premixing operations), its operation begins with a processing time of 100 minutes. After the reaction operation is completed the entire batch is transferred to a storage tank (T1). To differentiate the partial batches at the premixing operation from the final batch at the reactor operation, the latter is referred to as the output batch.

For product type 2, the raw materials are discharged from tanks A1 and A2, and stainless-steel or glass equipment types are acceptable for the premixing and reacting

operations. The reaction time for product type 2 is 60 minutes and the final product is stored in tank T2.

A *process plan* defines the set of equipment and junctions that satisfy the operation requirements specified in the recipe. It is common in multi-purpose plants to have alternative process plans that satisfy the operation requirements specified in the recipe. The sample recipes in Fig. 2 indicate that more than one piece of equipment can perform an operation. The process plan is represented by a set of *process chains*. A process chain is a path between any two equipment types and is specified by the machine at the start of the path, the junction, and the machine at the end of the path. The processing time of a process chain is for the operation processed by the machine at the start of the path.

Process chain	Processing time	Process chain	Processing time
Process plan 1:		*Process plan 2:*	
(A3, J1, M3)	0·0	(A3, J1, M4)	0·0
(A4, J1, M4)	0·0	(A4, J1, M3)	0·0
(M3, J3, R4)	10·0	(M4, J3, R4)	10·0
(M4, J3, R4)	5·0	(M3, J3, R4)	5·0
(R4, J4, T1)	100·0	(R4, J4, T1)	100·0

Table 2. Process plans for chemical type 1.

Process chain	Processing time	Process chain	Processing time
Process plan 1:		*Process plan 2:*	
(A1, J1, M1)	0·0	(A1, J1, M2)	0·0
(A2, J1, M2)	0·0	(A2, J1, M1)	0·0
(M1, J2, R1)	10·0	(M2, J2, R1)	10·0
(M2, J2, R1)	5·0	(M1, J2, R1)	5·0
(R1, J4, T2)	60·0	(R1, J4, T2)	60·0
Process plan 3:		*Process plan 4:*	
(A1, J1, M2)	0·0	(A1, J1, M1)	0·0
(A2, J1, M1)	0·0	(A2, J1, M2)	0·0
(M2, J2, R2)	10·0	(M1, J2, R2)	10·0
(M1, J2, R2)	5·0	(M2, J2, R2)	5·0
(R2, J4, T2)	60·0	(R2, J4, T2)	60·0
Process plan 5:		*Process plan 6:*	
(A1, J1, M3)	0·0	(A1, J1, M4)	0·0
(A2, J1, M4)	0·0	(A2, J1, M3)	0·0
(M3, J3, R3)	10·0	(M4, J3, R3)	10·0
(M4, J3, R3)	5·0	(M3, J3, R3)	5·0
(R3, J4, T2)	60·0	(R3, J4, T2)	60·0
Process plan 7:		*Process plan 8:*	
(A1, J1, M3)	0·0	(A1, J1, M4)	0·0
(A2, J1, M4)	0·0	(A2, J1, M3)	0·0
(M3, J3, R4)	10·0	(M4, J3, R4)	10·0
(M4, J3, R4)	5·0	(M3, J3, R4)	5·0
(R4, J4, T2)	60·0	(R4, J4, T2)	60·0

Table 3. Process plans for chemical type 2.

For example, the processing time of chain $(R4, J4, T1)$ is for the reaction operation performed by machine R4.

For sample product type 1, two process plans satisfy the operation requirements and are shown in Table 2. The two process plans differ only in the specification of which mixer performs the premixing operation. Note that none of the two process plans uses the stainless-steel mixers (M1 or M2) for the premixing operation because they are not connected to the glass reactor (R4). Table 3 shows the process plans that satisfy the recipe for product type 2. Eight feasible process plans (two for each reactor) exist for the manufacture of product type 2.

Figure 3 summarizes the technological configuration of the process. A solid line represents a membership link; for example, a plant may manufacture several product types. A dashed line represents a decisional link; for example, to manufacture a batch of a chemical, one selects a specific process plan. Not that a process chain can be in more than one process plan, and a process plan can constitute the manufacturing path or more than one product type.

In summary, the complete set of characteristics and assumptions of the batch chemical manufacturing environment that is modelled are as follows:

(1) An order specifies the product type, the order quantity, and the due date. A batch may satisfy the demand of more than one order of the same product type, or conversely, several batches may be required to meet the demand of a particular order. The order processing relationships are summarized in Fig. 4.

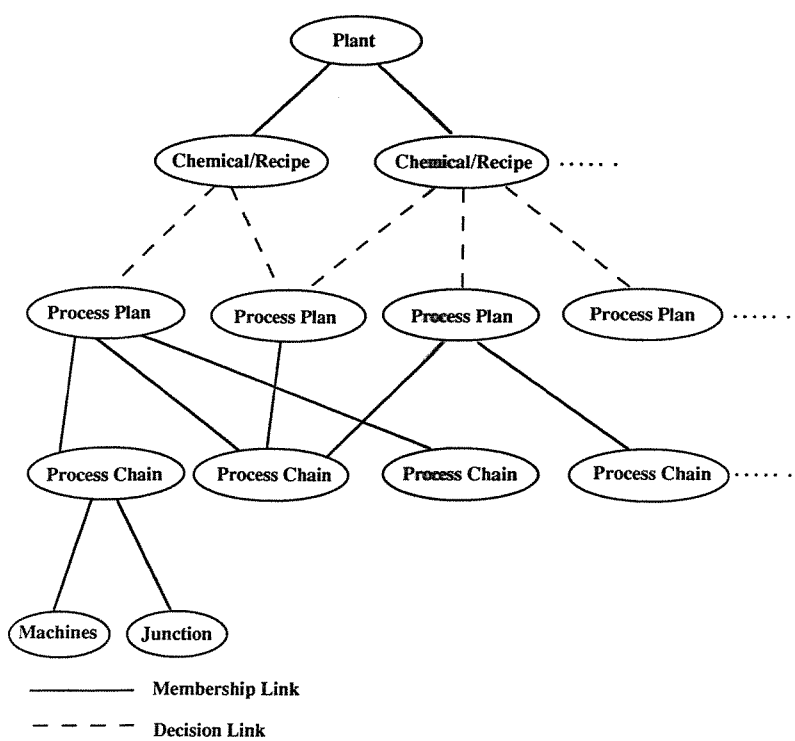

Figure 3. Technological configuration of the process.

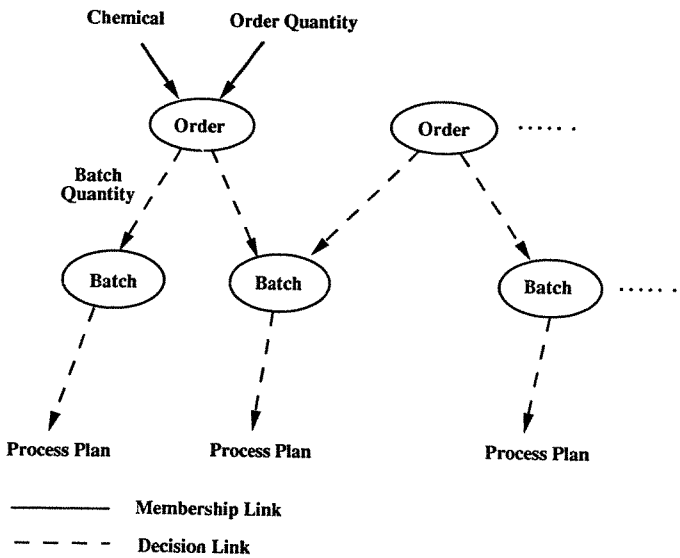

Figure 4. Order processing.

(2) The batch processing time is independent of the batch size. The transfer time is a function of the batch size and the transfer rate. The setup and cleaning times are assumed to be sequence independent and may be included in the batch processing time.
(3) All input data such as processing times, order quantities, and due dates are deterministic.
(4) Alternative process plans exist for each product type. The selected process plan is determined by the scheduling algorithm.
(5) Junctions must be used to connect process equipment and only one batch may be transferred through a junction at the same time. A batch can only be transferred if the piece of equipment used for the next operation is idle.
(6) Processing of a batch may not begin until the preceding operation for the batch is completed. A batch can wait indefinitely at its current operation if the succeeding operation is busy.

3. Notation

Before the model formulation is presented, the notation for the parameters and decision variables of the model is defined.

(1) *Indices*

i customer order index, $i = 1, \ldots, NO$, where NO is the number of orders.
b output batch index, $b = 1, \ldots, NB$, where NB is the number of output batches.
j product type index, $j = 1, \ldots, NC$, where NC is the number of different product types.
l machine index, $l = 1, \ldots, NM$, where NM is the number of machines.
u junction index, $u = 1, \ldots, NJ$, where NJ is the number of junctions.
k process plan index for product type j, $k = 1, \ldots, K_j$, where K_j is the number of different process plans for product j.
r process chain index, $r = 1, \ldots, NR$, where NR is the number of process chains.

(2) *Parameters*

h_i product type of order i.
d_i due date of order i.
q_i order quantity of order i (kgs).
c_l capacity of machine l (kgs).
f_l fraction of the output batch processed at machine l.
v_u transfer rate at junction u (kgs/min).
ms_r machine at start of process chain r, where machine ms_r performs the processing of the operation in process chain r.
μ_r junction in process chain r.
me_r machine at end of process chain r.
PC_r set of parameters of process chain r where $PC_r = (ms_r, \mu_r, me_r)$.
$a_{j,k,r}$ an indicator variable that equals 1 if process chain r is in process plan k for product j, and 0 otherwise.
$PP_{j,k}$ process plan k for product type j. It consists of a set of process chains expressed as:

$$PP_{j,k} = \left\{ \bigcup_{r=1,\ldots,NR} PC_r | a_{j,k,r} = 1 \right\}.$$

$t_{j,k,r}$ operation processing time of process chain r in process plan k for product j (min).

(3) *Computed variables*

H_b product type of output batch b.
CB_b completion time of output batch b (mins).
V_b size of output batch b (kgs).
$A_{i,b}$ amount of order i in output batch b (kgs).
CO_i completion time of order i (mins).
PL_b process plan used to manufacture batch b.
$SM_{b,l}$ start of processing time of batch b on machine l.
$SJ_{b,r}$ start of transfer time of batch b on junction in process chain r. The start of transfer time is indexed by the process chain index r instead of the junction index u because there can be multiple transfers of output batch b at junction u (i.e. batch merging).
$CM_{b,l}$ completion time of batch b on machine l. The batch is not considered complete from machine l until it is transferred out of machine l.
$CJ_{b,r}$ completion time of batch b on junction in process chain r.

Because of the possibility of alternative process plans and combining multiple orders in a single batch, the total number of batches (NB) required to meet the demand quantities is not known prior to determining the schedule. NB may be initially set to a sufficiently large number or to the upper bound on the number of batches required to meet the demand. It should also be noted that our formulation treats the variable f_l as a parameter, and in other applications f_l may be a decision variable.

4. Model formulation

Given the order quantities of all outstanding orders, it is desired for each batch b to determine the batch size (V_b), the product type of the batch (H_b), and the completion time of the batch (CB_b) to satisfy the demand under various constraints. The constraints and objective are as follows.

4.1. *Selection of process plan*

Each batch consists of only one product type and follows one predetermined process plan.

$$\sum_{j=1}^{NC} \sum_{k=1}^{K_j} z_{b,j,k} = 1 \quad \text{for } all\ b$$

where

$z_{b,j,k}$ a decision variable which takes on a value of 1 if batch b is of product type j and uses process plan k, and 0 otherwise.

Given $z_{b,j,k}$, PL_b can be defined as:

$$PL_b = PP_{j,k} \quad \text{such that } z_{b,j,k} = 1$$

4.2. *Restriction on batch size*

The batch size cannot be greater than the capacity of the equipment sets in the selected process plan.

$$z_{b,j,k} V_b \leq \frac{c_l}{f_l} \quad \text{for } all\ b, j, k, \text{ and } all\ l \in PP_{j,k}$$

To simplify the notation, the constraints are represented in terms of PL_b. The capacity constraint can be written as follows:

$$V_b \leq \frac{c_l}{f_l} \quad \text{for } all\ b \text{ and } l \in PL_b$$

4.3. *Satisfaction of demand*

The order quantity of each order i must be satisfied.

$$\sum_{b=1}^{NB} \sum_{k=1}^{K_{h_i}} z_{b,h_i,k} A_{i,b} = q_i \quad \text{for } all\ i$$

4.4. *Allocation of batch to orders*

The total allocation of the batch to orders must equal the batch size.

$$\sum_{i=1}^{NO} A_{i,b} = V_b \quad \text{for } all\ b$$

4.5. *Restriction on start of transfer time*

For all process chains in PL_b of batch b, the transfer to the machine at the end of the process chain cannot begin until the operation performed by the machine at the start of the process chain is completed.

$$SM_{b,l_s} + t_{b,r} \leq SJ_{b,r} \quad \text{for } all\ b,\ all\ (l_s, u, l_e) = PC_r \in PL_b$$

where

$$t_{b,r} = \sum_{j=1}^{NC} \sum_{k=1}^{K_j} t_{j,k,r} z_{b,j,k}$$

4.6. *Restriction on start of operation processing*

For all process chains in PL_b of batch b, the start of operation processing cannot begin until the transfer of the batch is completed.

$$CJ_{b,r} \leqslant SM_{b,l_e} \quad \text{for } all\ b,\ all\ (l_s, u, l_e) = PC_r \in PL_b$$

where

$$CM_{b,l_s} = CJ_{b,r} = SJ_{b,r} + \frac{V_b f_{l_s}}{v_u}$$

4.7. *Restriction on processing multiple batches on machine l at the same time*

Constraints are needed to prevent multiple batches from being processed on machine l at the same time. Let B_l be the set of all batches to be processed on machine l and $b \in B_l$ if batch b requires processing on machine l.

$$SJ_{b_2,r_2} \geq CJ_{b_1,r_1} - M(1 - y_{b_1,b_2,l}) \quad \text{for } all\ b_1,\ b_2 \in B_l \text{ and for } all$$

$$PC_{r_1} = \{(ms_r, \mu_r, me_r) | ms_r = l \text{ and } PC_r \in PL_{b_1}\}$$

and

$$PC_{r_2} = \{(ms_r, \mu_r, me_r) | me_r = l \text{ and } PC_r \in PL_{b_2}\}$$

$$SJ_{b_1,r_1} \geq CJ_{b_2,r_2} - My_{b_1,b_2,l} \quad \text{for } all\ b_1,\ b_2 \in B_l \text{ and for } all$$

$$PC_{r_2} = \{(ms_r, \mu_r, me_r) | ms_r = l \text{ and } PC_r \in PL_{b_2}\}$$

and

$$PC_{r_1} = \{(ms_r, \mu_r, me_r) | me_r = l \text{ and } PC_r \in PL_{b_1}\}$$

where

$y_{b_1,b_2,l}$ a decision variable which takes on a value of 1 if batch b_1 precedes batch b_2 (not necessarily immediately) for processing on machine l, and 0 otherwise.

M a very large number.

If batch b_1 precedes batch b_2 for processing on machine l, then the start of transfer of batch b_2 to machine l can begin only after batch b_1 has been transferred from machine l.

4.8. *Restriction on transferring multiple batches on junction u at the same time*

These constraints prevent multiple batches from being transferred on junction u at the same time. Let the set of all batches using junction u be G_u and $(b, r) \in G_u$ if batch b uses junction u in process chain r.

$$SJ_{b_2,r_2} \geqslant CJ_{b_1,r_1} - M(1 - x_{b_1,r_1,b_2,r_2}) \quad \text{for } all\ (b_1, r_1), (b_2, r_2) \in G_u$$

$$SJ_{b_1,r_1} \geqslant CJ_{b_2,r_2} - Mx_{b_1,r_1,b_2,r_2} \quad \text{for } all\ (b_1, r_1), (b_2, r_2) \in G_u$$

where

x_{b_1,r_1,b_2,r_2} a decision variable which takes on a value of 1 if batch b_1 on process chain r_1 precedes batch b_2 on process chain r_2 (not necessarily immediately) for transfer on junction u, and 0 otherwise.

If batch b_1 on process chain r_1 precedes batch b_2 on process chain r_2 for transfer on junction u, then the start of batch b_2 on process chain r_2 can begin only after batch b_1

on process chain r_1 has been transferred from junction u. In some cases, $b_1 = b_2$ since the same output batch may require multiple transfers from junction u (i.e. batch merging).

4.9. *Objective*

The proposed system develops schedules to minimize the total tardiness, formally stated as follows:

$$\min\left(\sum_{i=1}^{NO} \max\left(CO_i - d_i, 0\right)\right)$$

where

$$CO_i = \max_{b' = (b|A_{i,b} > 0)} (CB_{b'}) \quad \text{and} \quad CB_{b'} = \max_{l \in PL_{b'}} (CM_{b',l})$$

5. Scheduling algorithm

The problem described is similar to an N-job, M-machine job-shop scheduling problem in discrete-parts manufacturing with batch sizing, alternative routeing, zero buffering, and batch merging. As shown, the problem can be formulated as a mixed-integer non-linear model. There are no existing procedures capable of finding an optimal solution to this problem in a realistic amount of computational time. The job-shop scheduling problems without the additional constraints of this model have been shown to be NP-hard (Garey *et al.* 1976). For this reason, a heuristic-based solution procedure to minimize the total tardiness is proposed.

The algorithm schedules one order at a time. The order that has the least slack is scheduled, where the stack is defined to be the difference between the completion time of the order (CO_i) and the due date of the order (d_i). The least slack rule is used because previous studies show that this rule tends to perform well in minimizing the total tardiness in a job-shop environment (Morton and Pentico 1993).

The completion time of an order (CO_i) is a function of the selected processes used to manufacture the product type of the order. Refer to the set of processes that satisfy a particular order as a manufacturing plan. A manufacturing plan specifies the process plan used to produce each batch required to satisfy a particular order. As an example, consider an order of 4000 kgs for product type 2 shown in Fig. 2. As previously stated, there are eight process plans that satisfy the operation requirements of product type 2. Four of the process plans ($PP_{2,1}$, $PP_{2,2}$, $PP_{2,3}$, $PP_{2,4}$) have a capacity of 2000 kgs (the capacity of the process plan is the smallest capacity of any equipment in the process plan) and the other four process plans ($PP_{2,5}$, $PP_{2,6}$, $PP_{2,7}$, $PP_{2,8}$) have a capacity of 4000 kgs. If any one of the process plans that have a capacity of 2000 kgs is used to manufacture the chemical, then two batches are required to satisfy the order quantity. Therefore, there are 36 different manufacturing plans that satisfy the order quantity of 4000 kgs for product type 2. The manufacturing plan (1, 2) states that the first batch is produced using process plan $PP_{2,1}$ and the second batch is produced using process plan $PP_{2,2}$. Since the algorithm schedules one batch at a time and the same resource can be in multiple process plans, the completion time of the order using manufacturing plan (1, 2) can be different from using manufacturing plan (2, 1). The set of manufacturing plans for this example is:

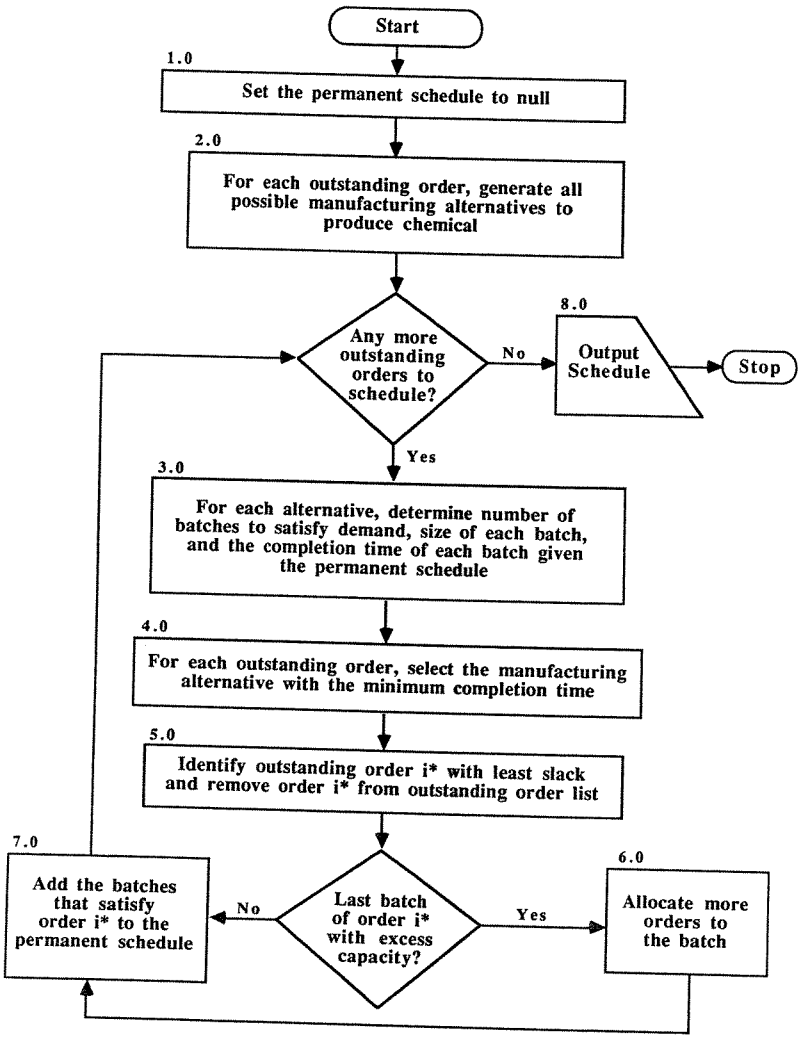

Figure 5. Outline of algorithm BATCH_SCHED.

(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (3, 8), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (4, 8), (5), (6), (7), (8)

For each order, the algorithm determines the number of batches required to satisfy the order quantity, the size of each batch, and the completion time of each batch on each resource in the selected process plan. Once the completion time for a particular batch has been determined, it remains fixed. That is, a later scheduled batch can be placed ahead of a previously scheduled batch in the sequence of a particular machine only if it does not delay the completion times of any previously scheduled batch.

The proposed scheduling algorithm is referred to as BATCH_SCHED and an overview is shown in Fig. 5. Step 1 of the algorithm places all the orders in an outstanding order list and initializes the permanent schedule to null. Step 2 generates all the manufacturing plans for each outstanding order i.

For each generated manufacturing plan of outstanding order i, Step 3 determines the size and the completion time of each batch in the manufacturing plan given the permanent schedule. To determine the completion times, Step 3 creates a temporary schedule of each batch on each resource in the process plan. For every machine l in the process plan to manufacture the batch, the procedure tries to find the earliest time in the permanent schedule of machine l where there is a sufficient difference in completion times of any two successive prior scheduled batches. In order for the difference in completion times (gap duration) to be sufficient, it must be large enough so as not to delay the completion time of any previously scheduled batch. Hence, the gap duration must be greater than the occupancy time of the batch at machine l and includes the transfer time of the batch of machine l, the operation processing time at machine l, the transfer time of the batch from machine l, any waiting time for the junction resources performing the transfer, and any waiting time for the machine performing the next operation. If no gap duration is sufficient, then the batch is placed at the end of the sequence for machine l.

After the completion times are calculated, Step 4 of the algorithm selects the best manufacturing plan to satisfy each outstanding order i. The manufacturing plans with the smallest completion time are identified to ensure that the order with the least slack finishes processing as soon as possible. If there is more than one manufacturing plan of order i that yields the minimum completion time, one of three tie breaking rules is employed. The first rule is to select the manufacturing plan with the minimum number of batches needed to satisfy the order. The intent is to minimize the amount of equipment used to manufacture the chemical on the order so the conflict with batches that have yet to be scheduled is minimized. The second rule is to select the manufacturing plan that does not contain a resource such that there are no other resources in the factory of the same type. This rule helps ensure the critical resources are free for other batches that may require its use. The final rule is to select the manufacturing plan with the most remaining capacity in the last batch of the manufacturing plan so that other orders requiring the same product type can be processed by this manufacturing plan.

Step 5 sets the earliest completion time of each outstanding order i to the maximum completion time of any batch in the selected manufacturing plan of the order. Then, the order (i^*) with the least slack is identified to be scheduled next (i.e. added to the permanent schedule) and is removed from the outstanding order list.

Note that the batch size of the last batch in the manufacturing plan required to satisfy order i^* can be below the smallest equipment capacity in its process plan. If this is the case, more orders are allocated to the last batch in the manufacturing plan (Step 6 is discussed in more detail below). Otherwise, Step 7 adds the schedule of batches that satisfy order i^* to the permanent schedule, and the algorithm returns to Step 3 to find another outstanding order to schedule. Finally, if there are no more orders to schedule, the algorithm outputs the schedule and stops (Step 10).

Step 6 involves identifying another outstanding order (i') requiring product type h_{i^*} to allocate to the last batch in the selected manufacturing plan of order i^*. Refer to this batch as batch b^l. The amount of order i' that is allocated to batch b^l (A_{i',b^l}) is restricted by the capacity of the bottleneck machine in the process plan used to manufacture batch b^l. Since allocating order i' increases the size of batch b^l and the transfer times are a function of the batch size, the completion time of batch b^l must be recomputed. If the allocation of order i' to batch (b^l) satisfies the demand of order i' (i.e. $A_{i',b^l} = q_{i'}$), then the algorithm sets the completion time of order i' to the completion time of batch b^l (i.e. $CO_{i'} = CB_{b^l}$), removes order i' from the outstanding order list, and attempts to

	Mean total tardiness		
Due dates	BATCH_SCHED	EDD	SOQ
Tight	54·5	84·3	112·7
Loose	0·2	27·7	43·0
Scattered	10·8	53·2	80·9

Table 4. Experimental comparison of heuristic rules.

identify another outstanding order to allocate to batch b^l. Otherwise, the manufacturing plans for order i' must be regenerated since the remaining order quantity has been reduced, and the algorithm goes to Step 7.

6. Example

This section compares the proposed scheduling algorithm BATCH_SCHED with several simple heuristic rules on the configuration shown in Fig. 1. Honeywell Industrial Automation and Control Division provided this configuration to test the proposed scheduling algorithm on actual data. This configuration is used to manufacture multiple chemical consumer goods. There are typically six outstanding orders at any given time consisting of three product types of equal product mix. Recipes for product types 1 and 2 are represented in Fig. 2. The recipe for product type 3 is identical to product type 2 except the final product is stored in tank T3. The rate of transfer is assumed to be 200 kgs/minute.

Algorithm BATCH_SCHED is compared with the earliest due date (EDD) and smallest order quantity (SOQ) heuristic rules based on the total tardiness criteria. The batches that consist of the smallest order quantities have small batch sizes if there is limited merging of orders. Although the operation processing times are independent of the batch sizes, the transfer times are a function of the batch size, and the rational behind the SOQ rule is to load the machines as quickly as possible.

For experimental purposes, the order quantity is sampled from a uniform distribution with a minimum of 3000 kgs and a maximum of 4000 kgs. The heuristics are compared using tight, loose, and scattered due dates. The tight due date scenario samples the due date for a particular order from a uniform distribution with a minimum of 240 minutes and a maximum of 300 minutes. The loose due date scenario samples the due date for a particular order from a uniform distribution with a minimum of 300 minutes and a maximum of 360 minutes. The scattered due date scenario samples the due date for a particular order from a uniform distribution with a minimum of 240 minutes and a maximum of 360 minutes.

The heuristic rules are tested on 100 randomly generated samples following the above distributions. Table 4 shows the average total tardiness of the 100 samples for each heuristic rule using the different due dates. The Newman–Keuls test (Anderson and McLean 1974) is used to perform a pairwise statistical test on the mean tardiness. All mean total tardiness are statistically significant at a confidence level of 99·9%. As the Table shows, algorithm BATCH_SCHED has the lowest mean tardiness in all the tested due date scenarios. Algorithm BATCH_SCHED performed well in terms of the total tardiness measure because the algorithm considers both the processing requirements and the due dates of the orders whereas the EDD rule only considers the due dates. The SOQ rule performed poorly because the order quantity is not a good

measure of the processing requirements of the orders and the rule does not consider the due dates of the orders.

These tests suggest that the proposed algorithm BATCH_SCHED outperforms the simple heuristic rules in minimizing the total tardiness on the sample configuration. However, further testing is required to generalize the results.

7. Implementation

The proposed scheduling system is developed in an object-oriented environment to naturally represent the system components and to allow rapid reconfiguration of the system components as conditions warrant. Suydam (1989), McCarthy (1990), and Dessouky *et al.* (1993) discuss the advantages of using an object-oriented environment to represent batch chemical processes. Smalltalk-80 is chosen as the development environment because of the existence of a complete set of predefined object libraries and availability on several computing platforms (RS 6000, 486 PC, Macintosh, SUN Stations, and DEC Stations).

The major benefit of an object-oriented approach is that the physical system components (i.e. machine and materials) and system characteristics (i.e. process plans and machine schedules) can be easily and naturally represented through the creation of objects emulating their structure and function. Figure 6 depicts the objects that are used by the scheduling system. Some of the objects in the scheduling system are order, product, recipe, equipment, schedule, and scheduling algorithm. An order object contains attributes that pertain to a customer order such as requested chemical, order quantity, due date, etc. A product object contains properties that define a unique type of chemical such as its heating coefficient, density, and reaction constant. A recipe object provides the manufacturing process steps for producing the chemical. An equipment object contains attributes such as capacity and equipment type specific to a piece of equipment in the process plant. The schedule object represents start and

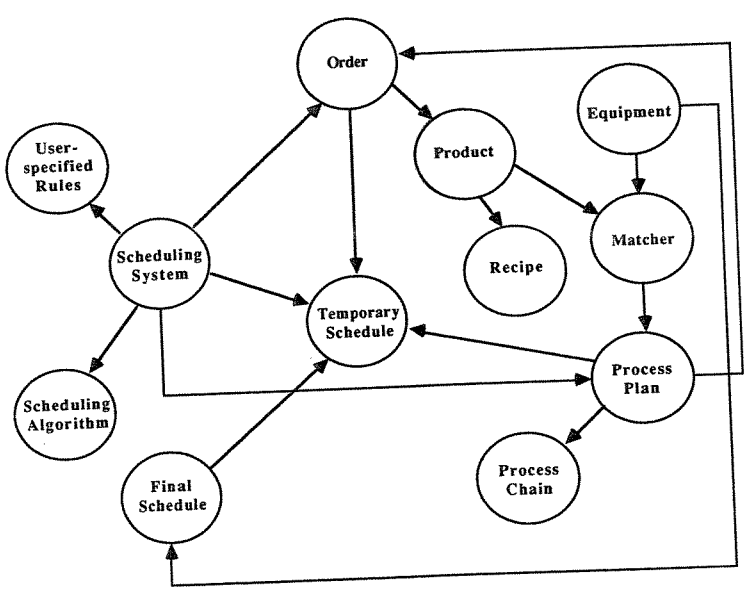

Figure 6. Overview of object interaction.

completion times of a batch on a piece of equipment. The scheduling algorithm object is used to derive the schedules.

As previously noted, there can be many different process plans that can manufacture a particular recipe. One role of the scheduling system is to identify all the available process plans for a particular recipe. The matcher object performs this functionality by identifying the equipment objects that can satisfy the recipe object for a particular product type. The process plan object is made up of process chain objects.

8. Conclusion and direction for future research

A mixed-integer non-linear model for the scheduling of general multi-purpose batch chemical plants is presented and a heuristic scheduling algorithm is developed to solve the model. The algorithm takes into consideration variable batch sizes and junction constraints. The algorithm also considers transfer times, alternative process plans, and batch merging. Experimental analysis of the algorithm BATCH_SCHED on an actual configuration showed that BATCH_SCHED outperformed simple heuristic rules in minimizing the total tardiness.

The algorithm schedules one batch at a time. Once the schedule for a particular batch has been determined, it remains fixed. Improvements to the overall schedule may be possible if adjustments to previous batch schedules can be made when determining the schedule of the current batch. For example, if the gap in the schedule of a particular machine is not sufficiently large to process the current batch, it might be better to push forward the start time of a previously scheduled batch instead of delaying the start time of the current batch.

Future research can also consider sequence dependent cleaning and setup times, storage tanks (buffer) in between equipment sets, and batch splitting and rejoining. In addition, some chemical properties such as molecular composition have stability time constraints after an operation. This time constraint influences the length of time a batch can be held for the availability of the header performing the transfer and the machine performing the successor operation.

References

ANDERSON, V. L., and MCLEAN, R. A., 1974, *Design of Experiments: A Realistic Approach* (New York: Marcel Dekker).

BIREWAR, D. B., and GROSSMAN, I. E., 1989, Efficient optimization algorithms for zero-wait scheduling of multiproduct batch plants. *Industrial and Chemical Engineering Research*, **28,** 1333–1345.

BIREWAR, D. B., and GROSSMAN, I. E., 1990, Simultaneous production planning and scheduling in multiproduct batch plants. *Industrial and Chemical Engineering Research*, **29,** 570–580.

DESSOUKY, Y., BEAUMARIAGE, T., ROBERTS, C., and OGLE, M., 1993, An intelligent system for batch scheduling for the process industry. *Computers in Industry*, **22,** 233–247.

EGLI, V. M., and RIPPIN, D. W., 1986, Short-term scheduling for multiproduct batch chemical plants. *Computers and Chemical Engineering*, **10,** 303–325.

GAREY, M. R., JOHNSON, D. S., and SETHI, R., 1976, The complexity of flowshop and job shop scheduling. *Mathematics of Operations Research*, **1,** 117–120.

KONDILI, E., PANTELIDES, C. C., and SARGENT, R. W., 1993, A general algorithm for short-term scheduling of batch operations—MILP formulation. *Computers and Chemical Engineering*, **17,** 211–244.

KU, H., and KARIMI, I., 1988, Scheduling in serial multiproduct batch processes with finite interstage storage: a mixed integer linear program formulation. *Industrial and Chemical Engineering Research*, **27,** 1840–1848.

KU, H., and KARIMI, I., 1990, Scheduling in serial multiproduct batch processes with due-date penalties. *Industrial and Chemical Engineering Research*, **29,** 580–590.

KU, H., and KARIMI, I., 1991, Scheduling algorithms for serial multiproduct batch processes with tardiness penalties. *Computers and Chemical Engineering*, **15,** 283–286.

KU, H., RAJOGOPALAN, D., and KARIMI, I., 1987, Scheduling in batch processes. *Chemical Engineering Progress*, **83,** 35–45.

MCCARTHY, J. J., 1990, The CIM reference model as a tool for plant information and control systems development. Technical report, Honeywell Inc. IACD, Phoenix, AZ.

MORTON, T. E., and PENTICO, D. W., 1993, *Heuristic Scheduling Systems* (New York: Wiley).

MUSIER, R. F., and EVANS, L. B., 1991, Schedule optimization with simultaneous lot sizing in chemical process plants. *American Institute of Chemical Engineers Journal*, **37,** 886–896.

PATSIDOU, E. P., and KANTOR, J. C., Scheduling of a multipurpose batch plant using a graphically derived mixed-integer program model. *Industrial and Chemical Engineering Research*, **30,** 1548–1561.

RICH, S. H., and PROKOPAKIS, G. J., 1986, Scheduling and sequencing of batch operations in a multipurpose plant. *Industrial Engineering Chemical Process Design and Development*, **25,** 979–987.

RIPPIN, D. W., 1983, Design and operation of multiproduct and multipurpose batch chemical plants—an analysis of problem structure. *Computers and Chemical Engineering*, **7,** 463–481.

RIPPIN, D. W., 1991, The big advantage offered by batch processing is flexibility. *Chemical Engineering*, May, 101–107.

SUHAMI, I., and MAH, R. S. H., 1982, Optimal design of multipurpose batch plants. *Industrial Engineering Chemical Process Design and Development*, **21,** 94–100.

SUYDAM, W., 1989, CASE makes strides toward automated software development. *Computer Design*, January.

TANDON, M., CUMMINGS, P. T., and LEVAN, M. D., 1991, Flowshop sequencing with non-permutation schedules. *Computers and Chemical Engineering*, **15,** 601–607.

VERHULST, J. D., 1989, Batch plant scheduling and coordination. *ISA Transactions*, **28,** 51–55.