A Continuous-Time Formulation for Scheduling Multi-Stage Multi-product Batch Plants with Identical Parallel Units

Liu Yu and I. A. Karimi
Department of Chemical & Biomolecular Engineering
National University of Singapore
4 Engineering Drive 4, Singapore 119260
cheiak@nus.edu.sg

Abstract

Multi-stage multi-product batch plants with parallel units are quite common in the batch chemical industry. How to assign products to those units optimally have become formidable scheduling problems considering the large number of assignment possibilities. To find a better way to solve these tough problems, we turn to find a way by studying the simplified problems (scheduling of multi-stage multi-product batch plants with identical parallel units) first. Instead of deciding the product sequence on each unit, we focus on the sequence of products at the start time and end time of each stage. Each stage is considered as a 'black box' in which we care about which time and who will enter and leave these 'black boxes' and need not care about how these 'black boxes' perform. We develop MILP formulations based on this idea. Examples show that formulations with this idea have a great reduction in binary variables and non-zero elements compared to the slot-based formulations existing in the literature. Finally, we find one formulation within the three formulations developed by us performs best comparing with the formulations in the literature.

Introduction

Multi-stage multi-product batch plants with parallel units are quite common in the batch chemical industry. In such plants, scheduling of plant operations is a routine activity. Due to the many alternate ways in which products can be assigned to various units and produced in different sequences, the task of optimal scheduling is formidable. Not much attention has been paid to scheduling in plants with multiple stages, in spite of its industrial significance. There are two kinds of representation (discrete-time and continuous-time) for these scheduling problems in the literature. Because discrete-time formulations often involve large numbers of binary variables, it often makes problems hard to solve. We consider the continuous-time representation here. A continuous time MILP model has been developed by Pinto and Grossmann (1995) to solve short term scheduling problems. In their model, sequence-independent set-ups and no resource constraints except equipment were the major assumptions. The problem has been modeled through a continuous time representation that relies on the time-slot notion and the use of parallel time axis for units and tasks. To assign orders to the specific slots of units in the stages, tetra-index binary variables, i.e. (order, slot, unit, stage) are used. Although these scheduling problems could be formulated clearly with this model, often a large number of binary variables will be introduced for larger problems. Ku and Karimi (1988) presented an optimal MILP formulation for scheduling N products across an M-stage serial processing system with single unit per stage. They considered the process under a combination of the intermediate storage policy as a combination of the UIS, FIS and UIS policies. To reduce the number of binary variables in the formulation, they developed a heuristic strategy of assigning products to specific neighborhoods of a sequence. They used minimizing makespan as the objective function. Gupta and Karimi (2003) developed a new MILP formulation for the short-term scheduling in a multistage batch plant with non-identical parallel units. They used a set of tri-index variables (order, order and stage) to handle order-sequence dependencies explicitly. By using this formulation , problems with both sequence-dependent and unit-dependent setup times could be solved. Comparing to the formulations in the literature, their new formulation could use fewer constraints and give better objective value than the previous works in less computational times. Model of Pinto and Grossmann (1995) belongs to slot-based models. Models of Gupta & Karimi (2003) belong to sequence-based models. We have compared these methods and our result shows that the slot-based models perform better than sequence-based models. However, formulations existing in the literature often involve large number of binary variables and cannot solve large problems.

To find a better way to solve these tough scheduling problems, we turn to find a way by studying the simplified problems first. Many multi-stage, multi-product non-continuous industrial plants use identical units in every stage. They are suitable for all the products and their processing rates are same. We choose these problems as the simplified problems. In this paper, we present three slot-based models to solve these simplified problems with UIS (Unlimited Intermediate Storage). For problems with UIS, there is no limitation for storage between stages and a batch can be held in its processing unit temporarily after its completion. Our key idea is as follows. Instead of deciding the product sequence on each unit, we focus on the start sequence and the end sequence of products in each stage. Each stage is considered as a ‘black box’ in which we care about which time and who will enter and leave these ‘black boxes’ and need not care about how these ‘black boxes’ perform. In this way, the unit assignment is avoided and the problem is simplified. This idea leads to a great reduction in binary variables and non-zero elements in our models compared to the slot-based models existing in the literature. We compare these models to model of Pinto and Grossmann (1995) and model of Gupta & Karimi (2003) and find the model, which performs best. Finally, this formulation can be extended to the general multi-stage multi-product problems.

Problem Description

Figure 1 shows a schematic diagram of a multi-stage multi-product simplified batch process.

The process comprises S stages with J parallel units (j =1, ..., J) and manufactures I distinct product items (i =1, ..., I). The units within a stage are identical which means that they are suitable for all the products and the processing times of a product on them are same. All the product items should be processed by a unit in a specific stage. For the convenience of formulation, we assume that there are two parallel identical units in every stage. By slight amendment, we can get formulations for other multistage problems with parallel (more than two parallel identical units) units easily. *We are going to extend our formulations to those problems later.* Intermediate storage policy such as UIS (Unlimited Intermediate Storage) is going to be considered.

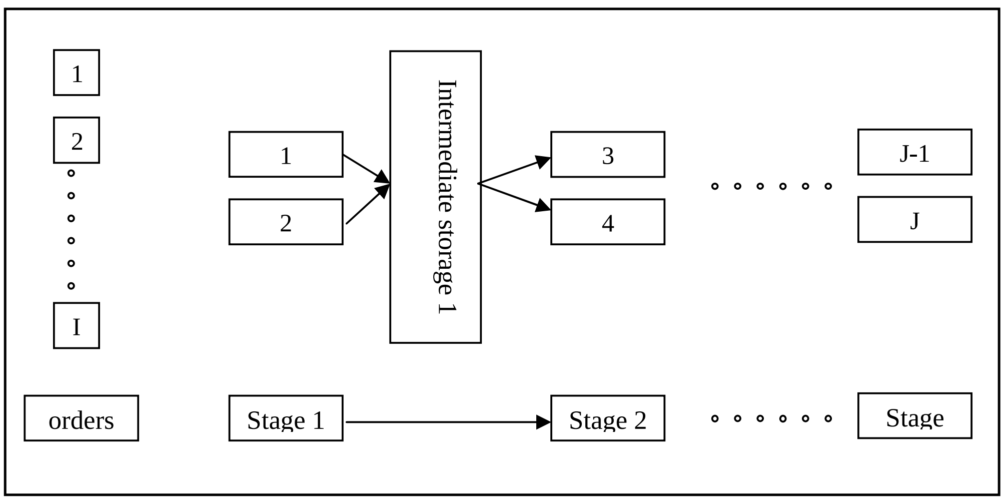

Figure 1. Schematic diagram of multi-stage multi-product process simplified batch process (with two identical units in each stage)

In addition to the above process features, we assume the following.

Assumptions

1. A unit cannot process more than one product at any time.
2. Processing is non-preemptive.
3. Processing units do not fail and processed batches are always satisfactory.
4. Time zero denotes the start of the current scheduling period.
5. Transition times are not considered.
6. Batch sizes are fixed parameters.

Each product is to be processed only once by exactly one unit of every stage it must go through.

There are different formulations to formulate any problem. For different intermediate storage polices, there are different formulations for the same problem. Here we are going to present three different formulations to solve these multi-stage multi-product simplified problems with the intermediate storage policy of UIS.

Model Formulations

Figure 2 shows the continuous-time representation in our models. In this time representation, we need consider the product sequence at start time and end time of each stage. The product sequence at start and end time of a specific stage is got by assigning the products to the start or end sequence slots the number of which is equal to the number of products. After the sequences have been decided, timing relationship among these slots can be got eventually.

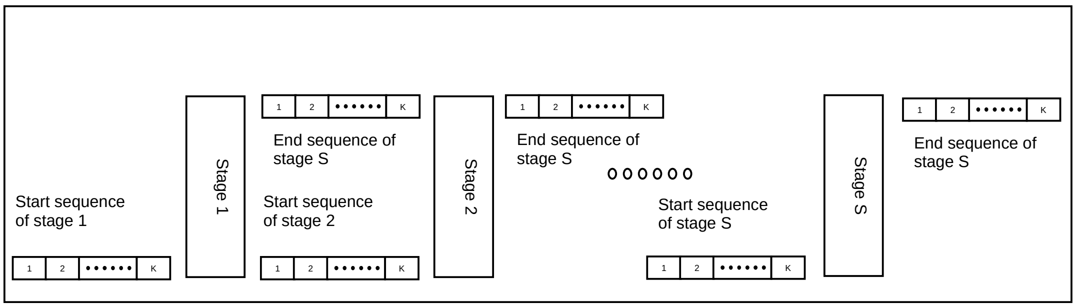

Figure 2. The continuous-time representation for the multi-stage multi-product simplified batch process

Mathematical Formulations

We construct three models based on the situation that there are two identical parallel units in each stage first. *Later we will give the extension models, which are suitable for the general problems with multiple identical units in each stage besides two identical units.*

Model 1

According to the above time representation, we develop the first model as following.

Allocation constraints

Binary *variables* YS_{iks} have been defined to assign product i to slot k at the start sequence for stage s.

$$YS_{iks} = \begin{cases} 1 & \text{if order } i \text{ has been assigned to slot } k \text{ at the start sequence for stage } s \\ 0 & \text{otherwise} \end{cases}$$

Binary variables YE_{iks} have been defined to assign product i to slot k at the end sequence for stage s.

$$YE_{iks} = \begin{cases} 1 & \text{if order } i \text{ has been assigned to slot } k \text{ at the end sequence for stage } s \\ 0 & \text{otherwise} \end{cases}$$

Allocation constraints

Constraints (1) show that each slot in the start sequence in a stage should process one product. Constraints (2) show that product i should start at a slot in the start sequence for a specific stage s .

$$\sum_i YS_{iks} = 1 \qquad \forall k, \forall s \qquad (1)$$

$$\sum_k YS_{iks} = 1 \qquad \forall i, \forall s \qquad (2)$$

Similarly, constraints (3) show that each slot in the end sequence in a stage should process one product. Constraints (4) show that product i should start at a slot in the end sequence for a specific stage s .

$$\sum_i YE_{iks} = 1 \qquad \forall k, \forall s \qquad (3)$$

$$\sum_k YE_{iks} = 1 \qquad \forall i, \forall s \qquad (4)$$

Constraints (5) show that the product, which ends at slot $k-1$ in the end sequence for stage s, must start at slot k in the start sequence for stage $s-1$ or those slots before it.

$$\sum_{1}^{k'=k} YS_{ik's} \geq YE_{i(k-1)s} \qquad \forall i, \forall s, k > 1 \qquad (5)$$

If product i starts at slot k, it should end at slot $k-1$ or the slots after it. Constraints (6) present this.

$$\sum_{k' \geq k-1}^{K} YE_{ik's} \geq YS_{iks} \qquad \forall i, \forall s, k > 1 \qquad (6)$$

Timing Constraints

Constraints (7) and (8) represent the time of slot k should be bigger than the time of slot $k-1$ in the start sequence and the end sequence correspondingly.

$$TS1_{ks} \geq TS1_{(k-1)s} \qquad \forall s, k > 1 \qquad (7)$$

$$TE1_{ks} \geq TE1_{(k-1)s} \qquad \forall s, k > 1 \qquad (8)$$

These constraints (9) show that slot k in the start sequence for stage s is not permitted to start until slot $k-2$ in the end sequence for stage s has ended. By doing so, we have made sure that at most two products are processing in each stage so that assigning products to units are avoided.

$$TS1_{ks} \geq TE1_{(k-2)s} \qquad \forall s, k > 2 \qquad (9)$$

Constraints (10) are the relationship of the start time of product i and its end time in stage s. Similarly, constraints (11) are the relationship of the start time of product i in stage s and its end time in stage $s-1$.

$$\sum_k TS2_{iks} + P_{is} \leq \sum_k TE2_{iks} \qquad \forall s, \forall i \qquad (10)$$

$$\sum_k TE2_{ik(s-1)} \leq \sum_k TS2_{iks} \qquad \forall i, s > 1 \qquad (11)$$

By using constraints (12) and (13), we could linearize the nonlinear constraints $TS2_{iks} = TS1_{ks} * YS_{iks}$. Similarly, we use constraints (14) and (15) to linearize the nonlinear constraints $TE2_{iks} = TE1_{ks} * YE_{iks}$

$$\sum_i TS2_{iks} = TS1_{ks} \qquad \forall k, \forall s \qquad (12)$$

$$TS2_{iks} \geq TS1_{ks} - M*(1 - YS_{iks}) \qquad \forall k, \forall s, \forall i \qquad (13)$$

$$\sum_i TE2_{iks} = TE1_{ks} \qquad \forall k, \forall s \qquad (14)$$

$$TE2_{iks} \geq TE1_{ks} - M*(1 - YE_{iks}) \qquad \forall k, \forall s, \forall i \qquad (15)$$

Objective function

We choose minimizing makespan as our objective. There are two kinds of expressions (constraints (16) and (17)) for our objective functions. Unlike the general formulations in the literature, we use these two sets of constraints in the same time to be our objective functions. In fact, we have discussed the advantage of using more than one set of constraints in the objective part in our previous work.

$$H = TE1_{ks} \qquad k \in K_l, s \in S_l \qquad (16)$$

$$H \geq \sum_k TE2_{iks} \qquad s \in S_l, \forall i \qquad (17)$$

Constraints (1)~(17) comprise model 1.

Model 2

Basing on the same allocation method in model 1, we have developed another two models. In model 2, we introduce another continuous positive variables TS_{is} (the start time of product i on stage s) instead of the continuous positive variables $TS2_{iks}$ and $TE2_{iks}$ in model 1 so that we can avoid using the summation parts ($\sum_k TS2_{iks}$ and $\sum_k TS2_{iks}$). Additionally, this leads to more big-M constraints.

Allocation constraints

The allocation constraints of this model are same as those of model 1 (constraints (1)~(6)).

Timing Constraints

We also use the timing constraints (7)~(9) of model 1 in this model. The other timing constraints are following.

Constraints (10) show the relationship of the start time of product i in stage s and its start time in the previous stage.

$$TS_{i(s-1)} + P_{i(s-1)} \leq TS_{is} \qquad s > 1, \forall i \qquad (18)$$

Constraints (11) and (12) keep the star time of product i in stage s is equal to the time of the slot k in start sequence when product i is assigned to slot k in stage s.

$$TS1_{ks} \geq TS_{is} - M*(1-YS_{iks}) \qquad \forall s, \forall i, \forall k \qquad (19)$$

$$TS_{is} \geq TS1_{ks} - M*(1-YS_{iks}) \qquad \forall s, \forall i, \forall k \qquad (20)$$

Constraints (13) represent the relationship of the product start time and the time of the slots in start sequence in stage.

$$TE1_{ks} \geq TS_{is} + P_{is} - M*(1-YE_{iks}) \qquad \forall s, \forall i, \forall k \qquad (21)$$

Constraints (14) represent the relationship of the product start time and the time of the slots in start sequence in the previous stage.

$$TS_{is} \geq TE1_{k(s-1)} - M*(1-YE_{ik(s-1)}) \qquad s > 1, \forall i, \forall k \qquad (22)$$

Objective function

Also, we choose minimizing makespan as our objective in model 2. We keep constraints (16) in this model and convert constraints (17) of model 1 to constraints (23).

$$H \geq TS_{is} + P_{is} \qquad s \in S_l, \forall i \qquad (23)$$

Till now, we have got model 2. Constraints (1)~(9) and (18)~(23) comprise model 2.

Model 3

In this model, we use two sets of positive continuous variables ($TE1_{ks}$ and TS_{is}) which are fewer than those of the previous two models.

Allocation constraints

Due to the same product assignment method as the two previous models, we use the same allocation constraints (constraints (1)~(6)) as those of the two previous models.

Timing Constraints

In this part, we also use constraints (8) to keep the time of end slot k bigger than that of end slot $k-1$.

Constraints (24) show that product i in stage s is not permitted to process until the slot $k-2$ of the end product sequence in stage s has ended if product i has been assigned to the slot k of the start product sequence in stage s. In fact, these constraints' function is same as that of those constraints (9) in the two previous models.

$$TS_{is} \geq TE1_{(k-2)s} - M*(1-YS_{iks}) \qquad \forall s, \forall i, k > 2 \tag{24}$$

Constraints (25) represent the relationship of the start time of product i in stage s and the time of slot k of the end product sequence in stage s.

$$TE1_{ks} \geq TS_{is} + P_{is} - M*(1-YE_{iks}) \qquad \forall s, \forall i, \forall k \tag{25}$$

Constraints (26) show that the relationship of the start time of product i in stage s and the time of slot k of the end product sequence in the previous stage.

$$TS_{is} \geq TE1_{k(s-1)} - M*(1-YE_{ik(s-1)}) \qquad s > 1, \forall i, \forall k \tag{26}$$

Constraints (27) represent the relationship of the start time of product i in stage s and the start time of product i in the previous stage.

$$TS_{i(s-1)} + P_{i(s-1)} \leq TS_{is} \qquad s > 1, \forall i \tag{27}$$

Objective function

Similarly, we use constraints (16) in this model also. Additionally, we choose constraints (28) as the other set in our objective part.

$$H \geq TS_{is} + P_{is} \qquad s \in S_l, \forall i \tag{28}$$

Consequently, constraints (1)~(6), (8), (16) and (24)~(28) comprise model 3.

Model Evaluation

To evaluate these three models and comparing them with the formulation developed by Pinto and Grossmann (1995) and the formulation by Gupta & Karimi (2003), we use the example developed by us with C language, which are new problems. This examples involve several stages (S) with some non-identical parallel units in each stage. All products pass through all stages in the same stage sequence. The following are the five examples.

Example: This example is concerned with the scheduling of 10 products in a two stages multi-product batch plant. There are 2 parallel identical units in the each stage. Data for example 1 are given in Table 1.

Table 1. Processing time of products in different stages

Product	S1	S2
O1	27	21
O2	20	24
O3	14	29
O4	28	28
O5	24	22
O6	22	30
O7	12	31
O8	19	20
O9	28	30
O10	22	20

We implemented all models in GAMS 20.7 and solved all problems with CPLEX 7.5 on an IBM notebook R40 running WINDOWS XP with single Intel Pentium processor 1.50 GHz having 384 RAM.

The solutions of this problem with our three models, the model of Pinto and Grossmann (1995) and the model of Gupta and Karimi (2003) are shown in Table 2.

Table 2. Solutions for models

Models	Time	Relative gap	Binary	Non-zero	RMIP	Absolute gap	Variables	Constraints
M1	5000	27.50%	400	4820	58	22	841	785
M2	398	0%	400	4610	58	0	461	1025
M3	5000	1.49%	400	3598	58	1	441	679
Gupta	5000	9.86%	300	7420	58	7	441	1652
Pinto	5000	20.55%	1320	12872	58	15	1601	2916

Even the numbers of binary variables of our models are more than that of the model of Gupta and Karimi (2003). According to the solutions, we can see model 2 can solve the problem with only 398 s while other models can not solve the problem within 5000 s. Also it seems that the performance of model 3 is better than that of model 1, the model of Pinto and Grossmann (1995) and the model of Gupta and Karimi (2003).

Conclusion

In this paper, a continuous time MILP formulation (model 2) for short-term scheduling of multi-stage multi-product batch plants with non-identical parallel units is proposed. Additionally, we have used one example to compare our models to the two models in the literature we mentioned before and shown that the performance of model 2 is best.