'91 KACC 1991. 10. 22~24

Near Optimal Production Scheduling for Multi-Unit Batch Process

Kyeong-Sook Kim　　Young-Sang Cho

Chemical Process Lab.

Korea Institute of Science and Technology

ABSTRACT

The determination of a production sequence is an important problem in a batch process operation. In this paper a new algorithm for a near optimal production sequence of N product in an M unit serial multiproduct batch process is proposed. The basic principle is the same as that of Johnson's algorithm for two-unit UIS system. Test results on a number of selected examples exhibit the superiority over previously reported results. In addition, a tabulation techniqe is presented to calculate the makespan of a given sequence of production for all processing units under UIS mode.

I. Introduction

In the 1950's, the mission of the chemical engineer was to transform old-fashioned batch processes into modern continuous ones. With such a perspective, it would be surprising to find that ,today, a significant proportion of the world's chemical production by volume and a much larger proportion by value is still made in batch processes and it does not seem likely that proportion will decline.

In general, the products produced by a continuous process have the characteristics of having a relatively simple reaction condition and being demanded for all year round. On the other hand, the products produced by a batch process have the characteristics of having a more complicated production sequence and being demanded occasionally or for special purposes. However in recent years as a result of higher income, consumers tend to prefer quality goods with delicate tastes. Thus the batch process which is suitable for the satisfying these requirement in many respects is gaining more attention.

To reflect these trends, a large number of new batch plants are already built or being built world-wide. And the completion among these companies are getting tougher and tougher. As a result, the optimization of batch processes becomes a major issue in chemical engineering.

Usually the optimization of processes, sometimes called the production planning, indicates a proper allocation of available resources such as processing time, equipment, inventories, labor and energy to meet market demands of products over an extended period of time in a most efficient or cost-effective manner. The sequencing and the detailed scheduling of process operations are thus subproblems within overall production planning. However if a proper production sequencing could be found to minimize the total time required to produce all the products ordered by customer, i.e. makespan, the overall production planning would be usually obtained. For this reason the most of articles published in this field have chosen the minimum makespan an objective function.

In this article, we propose a new algorithm to find a near optimal production sequencing with an objective function of the minimum makespan. In addition, we also suggest a table technique for the calculation of makespan under UIS policy.

The design problem without scheduling considerations, using a minimal capital cost design criterion was formulated by Ketner[1] and Loonkar and Robinson[2]. Sparrow et al.[3], they proposed a heuristic sizing procedure for multiproduct plants in the MULTIBATCH program. Grossmann and Sargent[4] used a nonlinear programming procedure for the design of multiproduct plant. Takamatsu et al.[5], Szwrac[6], Gupta[7] and Wiede et al.[8] presented comparative sudies of production planning for the multiproduct batch process with various storage policies. Ku and Karimi[9] proposed the algorithm for determination of completion time in serial multiproduct batch process. In an extension of this paper, Rajagopalan and Karimi[10] considered the same system with transfer and set-up times under the MIS policy. Ku and Karimi[11] and Yeh et al.[12] proposed the method for MILP (Mixed Integer Linear Programming) equation formulation. Recently, Malone[13] and Das et al.[14] suggested the simulated annealing method for the optimization of batch process.

2. Production Planning of multi-product batch process

Batch chemical plants are classified into two categories: multiproduct or flowshop and multipurpose or jobshop. In multiproduct plants, all products follow essentially the same sequence of processing units. In multipurpose plant, the products may follow different paths through the plant. This study focuses on the multiproduct batch plant.

Production planning of multiproduct plant, flowshop problem can be devided into two subproblems; sequencing and detailed scheduling. Sequencing involves determining the order in which the products are to be processed to obtain the best possible schedule. Detailed scheduling for a given sequence involves determining the starting and finishing times of each product on all processing units. While this is a trivial task in many plant configurations such as simplified flowshops, the problem is considerably more complicated in jobshops and complex flowshops.

2.1 previous algorithm for production planning

There are three common ways to solve the flowshop sequencing problem. The first method is utilizing heuristic algorithm. The second method reformulates the problem by using a Mixed Integer Linear Program (MILP) technique and solve it by commercially available optimization packages. The third method is to use the Branch and Bound (BAB) strategy.

For most flowshop sequencing problems, algorithms based on MILP or BAB will not be able to solve the problem with larger than about 12 products because of limitations. Thus a considerable amount of effort has been spent on developing suboptimal algorithms that are based on heuristics and finished in polynomial time. For the flowshop sequencing problems, these heuristic algorithms can be divided into two categories. Usually the first step of these algorithms generate a set of good sequences by heuristic reasoning and then with this initial sequence the method improves the sequence recursively until no improvement is possible. This algorithm gives superior solutions, but at the expence of more computation time. However, since it is not that expensive to implement improvement step with today's computing power, a recursive improvement phase is used in most heuristics. A common feature of most of the existing heuristic algorithms usually utilize exact results of Johnson's algorithm for two-unit flowshops, UIS (Unlimited Intermediate Storage) and ZW (Zero Wait)/NIS (No Intemediate Storage), in one way or another.

- Heuristic algorithm

Johnson's algorithm, the only classical but exact algorithm for a two-unit UIS system gives the sequence with the minimum makespan for all the two-unit UIS flowshop problem. Thus it is important to understand Johnson's algorithm. Johnson's algorithm can be summarized as follows.

1] Divide the N products into two sets denoted by P and Q such that all the products whose processing time on the second unit being longer than that on the first unit in P and the rest in Q.

2] Arrange the products in P with the ascending order of their processing times on the first unit.

3] Arrange the products in Q with the descending order of their processing times on the second unit.

4] Combine the sequences in P and Q, in that order, to get the optimal sequence with the minimum makespan.

For example, consider the following six-job and two-unit UIS flowshop :

Table 1. Example 2.1

Product	1	2	3	4	5	6
Processing Time						
on Unit 1	10	5	11	3	7	9
on Unit 2	4	7	9	8	10	15

For this problem, set P = [2,4,5,6] and set Q = [1,3], hence the optimal sequence is [4-2-5-6-3-1].

The simplest algorithm for a multi-unit problem, called Rapid Access with Extensive Search (RAES), was proposed by Dannenbring[15]. In spite of being one of the earliest algorithms of its kinds, RAES remains one of the most powerful technique. Its applicability is not restricted to the simplified UIS flowshop. It can be applied to more complicated flowshops. For example with some modifications it can be applied to the flowshops with transfer and set-up times. The first phase of RAES is the rapid access (RA) procedure that generates a initial sequence and the second phase is the extensive search (ES) procedure that recursively improves a current sequence.

Phase I : RA Step

1] Given an M-unit flowshop problem, generate a pseudo two-unit flowshop with the following artificial processing times.

$$a_j = \sum_{i=1}^{M} (M - j + 1) t_{ij} \qquad j = 1, 2, \ldots N \quad (1)$$

$$b_j = \sum_{i=1}^{M} i \cdot t_{ij} \qquad j = 1, 2, \ldots N \quad (2)$$

where ai and bi are the artificial processing times of product i on pseudo units 1 and 2 respectively.

2] Apply Johnson's rule to the above psedo two-unit problem to obtain a initial sequence.

Phase II : ES Proedure

1] Search the neighborhood of the current sequence for sequences with lower makespan. The neighborhood is defined as the N-1 sequences resulting from pairwise interchanges of adjacent products in the current sequence.

2] Select the sequence with the lowest makespan in the neighborhood as the new current sequence and repeat step 1. If the neighborhood does not contain any sequence with lower makespan, stop. The current sequence is the final solution.

The most important step here is the step for the calculation of completion time. In recent years, the algorithm, which Rajagopalan and Karimi[10] proposed, is used to calculate the makespan for a given sequence with different storage policies. The interstage storage policy determines the number of batches that can be temporarily held in storage between consecutive stages. Four different types of interstage storage policies are common in batch process:

1] Unlimited Intermediate Storage (UIS)
2] Finite Intermediate Storage (FIS)
3] No Intermediate Storage (NIS)
4] Zero Wait (ZW)

- MILP method

A MILP is an optimization problem whose objective function and constraints are all linear. It differs from a linear program (LP) in a sense that some of its variables are integer or binary (0-1). The binary varibles are defined as follows:

$$X_{ij} = \begin{cases} 1 : \text{if product } i \text{ is in position } j \text{ in production sequence} \\ 0 : \text{otherwise} \end{cases} \quad (3)$$

Since the criterion is to minimize the makespan, the objective function is to minimize C_{MN}, completion time of all products and constraints are as follows:

1] A product is assigned to one and only one position in the processing sequence.

2] A position in the sequence is assigned to one and only one product.

Under the above constraints, Completion time, C_{MN} is determined by following equation.

$$C_{ij} = MAX[C_{i(j-1)}, C_{(i+1)j}, C_{(i+1)(j-z_i-1)}] + t_{ij} \quad (4)$$

$$i = 1, 2, \ldots, M, \quad j = 1, 2, \ldots, N$$

where $C_{ij} = 0$, if $i \le 0$, $j \le 0$, or $i > M$.

The advatage in formulating the folwshop problem for an MILP is that commercial optimization packages such as LINDO(Linear, Interactive, Discrete, Optimizer), MPOS (Multipurpose Optimization System) etc. are widely available for solving MILP problems. In spite of its attractiveness, the MILP approach for the flowshop also has memory limitation as the size of a problem becomes larger.

Another way to solve the sequencing problem is to employ the BAB strategy. The procedure is basically an intelligent, partial enumeration of the set of all possible sequences. The basic idea is to branch the problem into the exclusive subproblem-all the possible permutation sequences in a sequencing problem can be branched into N subgroups with subgroup i, i=1,..N, containing all sequences starting with product i. The next step is to assign a lower bound to the objective function value that can be obtained from each subproblem. If a group has a lower bound that is greater than the current best objective value, we do not need to examine any sequence from that group since it cannot produce a sequence better than the current one. If a lower bound can be calculated with 100% guarantee, the final solution computed by the BAB algorithm is the optimum; otherwise, it may be suboptimal.

2.2 New algorithm for sequencing

The basic idea of the new algorithm proposed in this article is similar with that of Johnson's algorithm. But new algorithm can be applied to multi-unit flowshops with transfer times.

Let's consider the following simple scheduling problem.

Table 2. Scheduling example 2.2

product	1	2	3	4
unit 1	25	30	19	17
unit 2	10	15	21	23

If we chhoose a production sequence of [1-2-3-4], the makespan of this sequence for UIS policy can be calaulated by using the following tabulation technique.

Tabe 3. Makespan table for example 2.2

	production sequence				
unit	1	2	3	4	
α ..	25	55	74	91	
β ..		35	70	95	118 = makespan
	.	.	.	.	
	.	.	.	.	
	a	b	c	d	

The second row denoted by β shows the completion time of each product. For example, the completion time of product 1 is shown in the second column as 35. In this table, it is noted that the completion time of the product 2 is calculated by adding the processing time of product 2 in the second unit to the maximum value between the completion time of product 2 in the first unit and the completion time of product 1 in the second unit. That's to say that the completion time of product 2 in the third column of the second row is obtained by adding the maximum value between two numbers in the second row to the completion time of product 2 in the second unit.

In general, the makespan of this sequence can be calculated by using the following equation.

$$MS = T_1 + \sum_{i=2}^{N} L_i + \sum_{i=1}^{N-1} pos\ [\ F_{i+1} - (T_i - \sum_{k=1}^{i+1} F_k)\ ] \qquad (5)$$

where T_i is the total processing time of product i

L_i is the second unit processing time of product i

F_k is the first unit processing time of product i

MS is the makespan

* pos(a) = a if a > 0

= 0 if a ≤ 0

Thus for the example, makespan is as follows:

$$MS=35+59+pos(30-35+25)+pos(19-70-55)+pos(17-95+74)=118$$

The best sequence in this case can be obtained by applying the Johnson's rule as [4-3-2-1]. The makespan for this sequence can also be calculated by using equation (5) as follows.

$$MS=40+46+pos(19-40+17)+pos(30-61+36)+pos(25-81+66)=101$$

These results shows that equation (5) is exact mathematical representation to find makespan for a given sequence.

In equation (5), the first term represents the total processing time of the first product processed in the sequence, the second term represents the expected makespan for a given production sequence without any interruption, and the final term is the penalty imposed by the first unit processing delay.

It is noted here that the minimum value of the makespan can be obtained by just considering (T_i-ΣF_k) and F_{i+1} in the last term. In fact Johnson's rule provides the best way of sequencing by just looking at F_{i+1}'s and (T_i-ΣF_k)'s.

Now let's consider a more general scheduling problem as shown in below:

Table 4. Scheduling example 2.3

prod.	1		2		3	
unit	P	T	P	T	P	T
		5		6		3
1	2	3	4	3	7	4
2	1	2	2	7	3	9
3	4	1	3	6	2	1

where P denotes the processing time

T is the transfer time.

If we choose a production sequence of [1-2-3], the makespan of this sequence for UIS policy can be calculated by using the similar tabulation technique as in the previous example.

Table 5. Makespan table for example 2.3

unit	production sequence														
	1			2			3								
1	5	7	10	16	20	23	26	33	37						
2				10	11	13	23	25	32	37	40	49			
3							13	17	18	32	35	41	50	52	53

In this table, the three columns for each product represent the transfer time to a unit, the processing time in the corresponding unit, and the transfer time from the corresponding unit respectively. The number shows the accumulated completion time for the job.

If we assume that there is no delay upto the unit before the last unit, a similar mathematical representation to predict the makespan for the sequence of [1-2-3] can be formulated as follows.

$$MS = T_1 + \sum_{i=2}^{N} L_i + \sum_{i=1}^{N-1} pos\ [\ P_{i+1} - (T_i - \sum_{k=1}^{i+1} F_k\ )\ ] \qquad (6)$$

where T_i is the total processing time including the transfer time for product i

L_i is the last unit staying time including the transfer time for product i

F_k is the first unit staying time including the transfer time for product k

P_i is the processing time upto the unit before the last unit excluding the transfer time to the last unit for product i

* pos(a) = a if a > 0

= 0 if a ≤ 0

Equation (6) is similar in form to equation (5). The only difference between equation (5) and equation (6) is the term P_{i+1} in equation (6) instead of F_{i+1} in equation (5). Recalling that the best sequence for equation (1) can be found by applying Johnson's rule to simple two unit scheduling problems which consist of F_{i+1}'s and (T_i-F_i)'s, this indicates that the best solution for equation (6) can also be obtained by applying Johnson's rule to the psedo two-unit problems which consist of P_i's and (T_i-F_i)'s.

This logic provides us a tool to estimate a initial best possible sequence for the general scheduling problems. The procedure to find this sequence can be summerized as follows:

1] Calculate P_i's for every product i. (i = 1...N)

2] Calculate (T_i-F_i)'s for every product i. (i = 1...N)

3] Formulate psedo two-unit problem assigning P_i to the processing time for product i in the first unit, and (T_i-F_i) to the processing time for product i in the second unit.

4] Apply Johnson's rule to find initial sequence.

For example, applying the above procedure to the example 2.3.

1] $P_1 = 11$, $P_2 = 15$, $P_3 = 17$

2] $T_1 - F_1 = 8$, $T_2 - F_2 = 18$, $T_3 - F_3 = 15$

Thus the psedo two-unit problem for the example as follows:

Table 6. Modified example 2.3

	1	2	3
1	11	15	17
2	8	18	15

Applying Johnson's algorithm suggests that the initial best sequence should be [2-3-1]. Calculating the makespan for this sequence using tabulation technique gives the following results.

Table 7. Makespan table for sequence [2-3-1]

unit	production sequence														
	1			2			3								
1	6	10	13	16	23	27	32	34	37						
2				13	15	22	27	30	39	42	43	45			
3							22	25	31	40	42	43	45	49	50

makespan = 50

Since there is no extra delay in the final unit makespan, the same results can also be obtained by using equation (6) as follows:

$$MS = T_1 + L_2 + L_3 + pos(P_2 - T_1 + \sum_{k=1}^{2} F_k) + pos(P_3 - T_2 + \sum_{k=2}^{3} F_k)$$

$$= 31+7+12+pos(30-31)+pos(43-43) = 50$$

3. Examples and Results

4-3c

	1		2		3		4	
	P	T	P	T	P	T	P	T
		1		5		2		5
1	2	4	4	9	1	2	1	4
2	5	3	2	8	4	9	3	3
3	2	7	3	2	3	2	2	1

NA : 3-2-1-4 [62]
RAES : 4-3-2-1 [67]
* NA = New Algorithm

6-7a

	1		2		3		4		5		6	
	P	T	P	T	P	T	P	T	P	T	P	T
		1		3		3		4		2		1
1	1	2	4	2	1	2	3	6	1	5	3	1
2	2	3	8	1	5	7	2	3	2	4	4	2
3	6	4	2	8	4	3	2	7	7	7	5	3
4	3	2	3	3	3	8	8	6	8	2	2	4
5	2	4	1	5	2	2	5	6	6	5	8	1
6	8	2	1	2	6	5	2	2	3	1	6	2
7	6	1	7	4	1	4	4	3	2	4	3	4

NA : 1-6-2-3-5-4 [125]
RAES : 3-1-4-5-6-2 [135]

Table 8. Results for examples

	NA	NA - 1st	RAES	RAES - 1st	
3-3 a	52	52	53	52	o
b	50	50	57	54	o
c	54	54	56	54	o
3-4 a	43	43	43	43	△
b	52	52	52	52	△
c	67	67	69	69	o
3-5 a	63	63	63	63	△
b	71	70	73	71	o
c	67	67	76	67	o
4-3 a	73	67	75	71	o
b	64	62	68	63	o
c	62	62	67	64	o
4-4 a	77	76	77	77	o
b	30	30	30	30	△
c	64	64	66	64	o
4-5 a	78	78	82	78	o
b	84	84	88	84	o
c	102	96	109	108	o
5-5 a	75	74	76	74	o
b	105	101	98	98	x
c	110	104	110	106	o
5-6 a	110	107	108	107	x
b	111	108	107	107	x
c	91	88	94	87	o
6-6 a	107	104	110	108	o
b	118	118	123	116	o
6-7 a	125	124	135	124	o
b	165	159	154	151	x
8-5 a	137	131	142	132	o
b	130	128	129	129	x
9-9 a	201	187	188	180	x
b	221	201	228	215	o
10-10a	241	241	424	424	o
b	238	238	535	535	o

-1st : result after the first ES procedure
o : NA is superior to RAES
△ : NA is equivalent to RAES
x : NA is inferior to RAES

4. Conclusion

Production planning is of an immense importance in batch processes in CPI. The optimization of a production sequence or scheduling can significantly improve the productivity and cost effectiveness in noncontinuous processes. In this study, a new and efficient algorithm for finding a near optimal multi-unit production sequence is proposed and a basic idea of the simple table technique for determination of completion times for a given product sequence under UIS policy was also suggested. A research for the more flexible use of the table technique is proceeding now.

Many examples are tested for the comparison purpose between RAES algorithm and the new algorithm proposed in this work. Results show that the new algorithm is more effective than RAES algorithm as shown in Table 8. When a number of product is less than 6, the new algorithm is overwhelmingly effective and when a number of product is much larger than 10, the new algorithm can find a near optimal sequence while RAES algorithm fails to find a sequence with the same magnitude.

In conclusion the method proposed here is superior to RAES especially for the problem with more than 10 product. The tabulation technique to calculate makespan developed in this work is simpler than other methods and we hope to give ore insights in analyzing the batch process operation. By utilizing the new algorithm as invaluable guides in our continued work towards the solution of the general batch production planning problem, we hope we could report a lot more in near future.

References

1. S.E. Ketner," Minimize batch equipment cost," Chem. Engng, 67, pp.121-124, 1960.
2. Y.R. Loonkar and J.D. Robinson," Minimization of capital investment for batch processs," Ind. Engng Chem.,Process Des. Dev. 9,p.625, 1970.
3. R.E. Sparrow, G.J. Forder and D.W.Rippin," The choice of equipment sizes for multi-product batch plants," The Chem. Engng, 289,p.520, 1975.
4. I.E. Grossmann and R.W.H. Sargent," Optimal design of multipurpose batch chemical plants," Ind. Engng Chem., Process Des. Dev. 18,p.343, 1979.
5. T. Takamatsu, I. Hishimoto," Optimal design operation of a batch process with intermediate storage tanks," Ind. Engng Chem., Process Des. Dev. 21, p.431, 1982.
6. W. Szwrac," Flowshop problem with time lag," Mgmt Sci., 29,p.477, 1983.
7. J.N.D. Gupta," Flowshop schedules with sequence dependent set-up times," J.Opl.Res.Soc., Japan 29, p.206, 1986.
8. W. Wiede and G.V. Reklaitis," Determination of completion times for serial multiproduct processes," Comput. Chem. Engng, 1(4), pp.345-356, 1987.
9. H. M. Ku , D. Rajagopalan and I.Karimi," Scheduling in batch processes," Chem. Engng Prog., pp.35-45, August, 1987.
10. D. Rajagopalan and I. Karimi," Completion times in serial mixed-storage multi-product processes with transfer and set-up times," Comput. Chem. Engng, 13(1/2), pp.175-186, 1989.
11. H. M. Ku and I. Karimi," Scheduling in serial multiproduct batch processes with finite interstage storage: A MILP formulation," Ind. Eng. Chem. Res. , 27, pp.1840-1848, 1988.
12. N.C. Yeh and G.V. Reklaitis," Synthesis and sizing of batch/semicontinuous processes: single product plants," Comput. Chem. Engng, 11(6), pp.639-654, 1987.
13. M.F.Malone," Batch sequencing by simulated annealing," AIChE Annual Meeting, San Francisco, CA, paper 23f, 1989.
14. H.Das, P.T.Cummings and M.D. LeVan," Scheduling of serial multiproduct batch processes via simulated annealing," Comput. Chem. Engng, 14(12), pp.1351-1362, 1990.
15. D. G. Dannenbring, " An evaluation of flowshop sequencing heuristics," Manag.Sci., 23,p.1174, 1977.