Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 8 (11): 2136-2141, 2008
ISSN 1812-5654

Scheduling Three Stage Flowshop Processes with No Intermediate Storage Using Novel Matrix Approach

Amir Shafeeq, M.I. Abdul Mutalib, K.A. Amminudin and Ayyaz Muhammad
Department of Chemical Engineering, Universiti Teknologi Petronas, 31750 Tronoh, Perak Darul Ridzuan, Malaysia

Abstract: Scheduling optimisation normally aimed at minimizing makespan, leading to overall optimisation of the production cost. This study focuses on determining the production sequence with minimum makespan for scheduling of a flowshop process using no intermediate storage (NIS) transfer policy. Unlike previous methods which uses mathematics considerably and lacks interactivity, the method introduces a simple approach based on matrix formulation to determine the makespan for all possible batch production sequences thus allowing simple screening to be done for selecting the optimal sequence.

Key words: Batch process, scheduling, matrix, no intermediate storage

INTRODUCTION

The chemical industry has become more attracted towards batch processes due to their flexibility and suitability for the production of relatively small volume and high variety products (specialty chemicals, pharmaceuticals and agrochemicals), which offer advantages in present economic and business situations. The manufacturing of such products generally involves multi-stage unit operations with tight specifications. The significant development in flexible production systems has made it possible to produce wide variety of products using flowshop processes where more than one product is produced in a pre-specified sequence using same equipments in successive campaigns. More recent works have considered the more complicated cases of jobshop processes where each product has its own production sequence and make use of processing units in different combinations. As a result, several products may be produced concurrently (Voudouris and Grossmann, 1992; Orçun *et al*., 2001).

The competition among different products in a flowshop process for shared and limited production resources has caused complexity in scheduling problems. Much of the difficulty arises due to a large set of diverse constraints and multiple objectives that might be conflicting at times. Significant amount of works on mathematical programming approach has been applied to batch process scheduling by formulating the problem as mixed-integer linear or nonlinear problem. However, many industries are still hesitant to apply the scheduling techniques and mostly prefer to do so manually by their plant operators though the mathematical methods were made available through computer applications.

The problem with scheduling design of batch processes could be divided into different categories depending on the way the intermediate products are handled in between the unit operations. Such problems are normally referred to as applying different transfer policies. The scheduling problem will become more complicated when feed uncertainty and product market demand pattern are taken into account. On the transfer policy, cases that are usually considered consist of (i) Zero Wait (ZW), (ii) Unlimited Intermediate Storages (UIS), (iii) Finite Intermediate Storage (FIS) and (iv) No Intermediate Storage (NIS) Transfer Policy (Birewar and Grossmann, 1989; Das *et al*., 1990; Grau *et al*., 1996; Biegler *et al*., 1997). When different transfer policy is selected for a batch process operation, it has an impact on the optimum production sequence which, in turn affects the optimum makespan of the batch process. Generally, for a given batch process recipe, the problem refers to the determination of the makespan for corresponding optimum production sequence (Jung *et al*., 1994; Balasubramanian and Grossmann, 2002; Ryu and Pistikopoulos, 2007).

Contrary to the mathematical programming approach, works related to the use of heuristic approach for determining optimal batch process or production sequence were generally based on the use of Gantt charts which is simple but can be extremely tedious particularly for the case of large scale problems. Comparisons have to be made on huge number of different possible batch production sequences in order to find the optimal

Corresponding Author: Amir Shafeeq, Department of Chemical Engineering, Universiti Teknologi Petronas, 31750 Tronoh, Perak Darul Ridzuan, Malaysia Tel: +60124069384 Fax: +6053654090

solution. Nevertheless, it keeps the designer fully in control in terms of the selection of the batch production sequence during design process.

In this study, a simple matrix approach is proposed, which could calculate quickly the makespan of a specified batch production sequence from a given batch process recipes based on the No Intermediate Storage (NIS) transfer policy (Suhami and Mah, 1981; Ku and Karimi, 1988, 1990, 1992; Kim *et al*., 1996). The generic matrix formulation, could also be used for determining the makespan for all other possible production sequences that could be derived from the same batch process recipes. A list of possible production sequences together with their respective makespan can be generated from which screening can then be done to select a few for short listing as candidates for the optimal solution(s). The proposed matrix approach is also capable of handling relatively large number of products and with computer programming, the task of finding the optimal solution(s) could be achieved within a short span of time.

Formulation of batch scheduling problem using Gantt chart method: The basis for formulating solution strategy for any problem depends on having pertinent information and understanding the nature of the problem to be solved. In the case of batch scheduling design, these information consists of (i) the number of products to be produced, (ii) the number of process stages, (iii) the batch process recipe for each product and (iv) the interstage transfer policy used e.g., NIS in the present study.

The batch scheduling problem consist of three main stages i.e., mixing (S_1), reaction (S_2) and separation (S_3), with negligible clean up and transfer times is used for the purpose of elaborating the matrix approach.

Example 1: Makespan for two products i.e., AB: The processing time for two products (A and B) using three stages (S_1, S_2 and S_3) are shown in Table 1. The makespan of the process for products A and B is determined using Gantt chart method, which is 21 h shown in Fig. 1.

It is obvious that there are three possible paths to determine the makespan. Firstly, the makespan is calculated based on taking the sum of AS_1, BS_1, BS_2, BS_3 and the waiting period of the intermediate product in stage BS_2 i.e., 2 h as represented by the shaded area in Fig. 1. Secondly, it can also be calculated by taking the sum of AS_1, AS_2, BS_2, BS_3, the idle time after AS_2 i.e., 1 h (Fig. 1) and the waiting period in BS_2 i.e., 2 h (Fig. 1). Finally, it could also be calculated by taking the sum of AS_1, AS_2, AS_3 and BS_3. It is observed that result of makespan calculation is same from all the possible paths i.e., 21 h. Apparently, it seems to be a lot easier to use the last path, which does not require calculation of any idle or waiting time. However, this may not be so easy to choose when dealing with number of products greater than two as will be illustrated in the forthcoming examples.

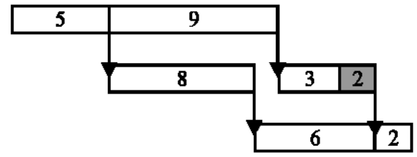

Fig. 1: Gantt chart for two products in three stages

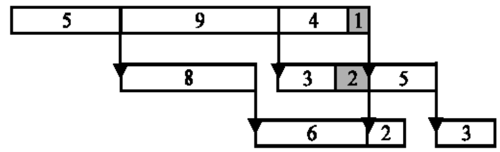

Fig. 2: Gantt chart for three products in three stages

Table 1: Processing time of two products in three stages

Products	Processing time (h) S_1	S_2	S_3
A	5	8	9
B	9	3	2

Table 2: Processing time of three products in three stages

Products	Processing time (h) S_1	S_2	S_3
A	5	8	6
B	9	3	2
C	4	5	3

Example 2: Makespan for three products i.e., ABC: In this example, the makespan calculation is performed for three products i.e., A, B and C. The respective processing time for the batch process recipes producing three products in the sequence of product A, followed by product B and lastly product C, are shown in Table 2. It appears from the Fig. 2 that possible number of paths to calculate makespan is now eight instead of three as earlier observed in example 1. The calculated makespan for the specified production sequence in this case is 27 h.

The first path to calculate makespan is by taking sum of AS_1, AS_2, AS_3, BS_3, idle time between BS_3 and CS_3 and CS_3. The second path is by taking sum of AS_1, AS_2, AS_3, CS_2 and CS_3. The third path is by taking sum of AS_1, AS_2, idle time between AS_2 and BS_2, BS_2, waiting time in BS_2, CS_2 and CS_3. The fourth path is by taking sum of AS_1, AS_2, idle time between AS_2 and BS_2, BS_2, waiting time in BS_2, BS_3, idle time between BS_3 and CS_3 and CS_3. The fifth path is by taking sum of AS_1, BS_1, BS_2, waiting time in BS_2,

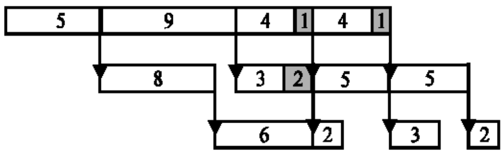

Fig. 3: Gantt chart for four products in three stages

Table 3: Processing time of four products in three stages

Products	Processing time (h) S_1	S_2	S_3
A	5	8	6
B	9	3	2
C	4	5	3
D	4	5	2

CS_2 and CS_3. The sixth path is by taking sum of AS_1, BS_1, BS_2, waiting time in BS_2, BS_3, idle time between BS_3 and CS_3 and CS_3. The seventh path is by taking sum of AS_1, BS_1, CS_1, waiting time in CS_1, BS_3, idle time between BS_3 and CS_3 and CS_3. The eighth path is by taking sum of AS_1, BS_1, CS_1, waiting time in CS_1, CS_2 and CS_3. Again, the makespan calculated from all the possible paths is same i.e., 27 h.

Example 3: Makespan for four products i.e., ABCD: In this example, the makespan calculation is performed for a batch process producing four products in the sequence of product A, followed by product B, then product C and finally product D, using three processing stages. The batch process recipes for the production of the four products are shown in Table 3. It is obvious from Fig. 3 that the number of possible paths to calculate makespan is increased further with increase in number of products as observed earlier in example 2 above. However, the result of makespan calculation is same from all possible paths i.e., 31 h.

From the above observations, it is concluded that number of paths to calculate makespan increases with increase in number of products and also depend on the batch process recipes i.e. processing time of each product in each stage. But the result of makespan calculation remains same from all the possible paths. This offers a flexibility to choose any path. However, it is suggested to use common path for any number of products for the purpose of developing a calculation procedure for makespan. From the observations made from example 1, 2 and 3, the path which offers makespan calculation by taking sum of processing times in the second stage of all products, idle and waiting times in second stages, processing time of first product in first stage and processing time of last product in last stage, could be selected as a common path for any number of products. The matrix approach is designed on the basis of this observation to calculate the makespan for any number of products.

THE PROPOSED MATRIX APPROACH

The past application of matrix has managed to simplify considerably the amount of calculation required for solving set of mathematical equations developed to represent a specific system. The proposed matrix formulation in the current work is also capable of simplifying the calculation required to determine the makespan for a specified batch production sequence as opposed to the Gantt chart method. The ability to quickly do so enables the matrix approach to be used to calculate the makespan for all possible production sequences derived from a given batch process recipes. A simple screening procedure could then be employed to select the few best sequences as potential solutions and the decision for the best option is left to the designer. The proposed matrix formulation does not involve any complex mathematical theorem but limited to only simple mathematical formulae combined with some logical understandings derived from the Gantt chart method. Also, it could be easily programmed on computer thus making the execution of the optimization procedure significantly faster and easier.

The procedure for the matrix approach: The proposed matrix approach was derived from observations made when applying the Gantt chart method to determine makespan of a specified batch production sequence. The idea is to replace the traditional Gantt chart method with an easier one that could calculate the makespan much quickly. The guideline developed for the matrix approach to calculate the makespan is provided below using the case of a batch process producing four products using three processing stages.

Step 1: Firstly, arrange the product recipes according to the arrangement shown below in the form of Matrix M_{ij} where A, B, C and D represent the products in rows i where i = 1, 2, 3, 4 while S_1, S_2 and S_3 represent the corresponding stages in column j where j = 1, 2, 3. In this respect the scheduling would be based on a sequence where product A is produced first, followed by product B, then product C and lastly product D.

	1	2	3
1	AS_1	AS_2	AS_3
2	BS_1	BS_2	BS_3
3	CS_1	CS_2	CS_3
4	DS_1	DS_2	DS_3

Step 2: On the basis of the observations made from above examples for the selection of common path to calculate makespan, select the first element in the first row, the entire elements in the second column and the third element in the bottom most row of the matrix i.e., in the above case the elements are AS_1, AS_2, BS_2, CS_2, DS_2 and DS_3.

	1	2	3
1	$\boxed{AS_1}$	AS_2	AS_3
2	BS_1	BS_2	BS_3
3	CS_1	CS_2	CS_3
4	DS_1	DS_2	DS_3

Step 3: It is also very much obvious from above examples that two parameters which are in addition to processing times in the stages, must also be included in makespan calculation namely (i) the idle time that exists between stages and it represents the delay prior to starting up the next stage and (ii) the waiting time required within the stage and it represents the waiting time for the intermediate products until the availability of the next stage. The calculation of idle time and waiting times can be done by introducing slack variables in between each of the second column elements in the matrix. The value of each slack variable will be based on summation of idle and waiting times calculated in between the second stage of all the products. In the present case, there will be three slack variables i.e., V_1 located in between elements AS_2 and BS_2, V_2 located in between elements BS_2 and CS_2 and V_3 located in between elements CS_2 and DS_2. Note that the number of slack variables will be one less, than the number of products. The respective idle times and waiting times are represented by letters d and w, respectively as illustrated below.

	1	2	3
1	AS_1	AS_2	AS_3
		V_1 (d_1+w_1)	
2	BS_1	BS_2	BS_3
		V_2 (d_2+w_2)	
3	CS_1	CS_2	CS_3
		V_3 (d_3+w_3)	
4	DS_1	DS_2	DS_3

First, the calculation of the slack variable is made based on the value of the matrix elements located diagonally between the first two rows i.e., BS_1 and AS_2 and BS_2 and AS_3.

For the matrix shown above the formula for calculating the slack variable V_1 between AS_2 and BS_2 is:

$V_1 = d_1 + w_1$ where $d_1 = BS_1 - (AS_2 + w_0)$ and $w_1 = AS_3 - (BS_2 + d_1)$

The value of w_0 is assumed to be zero at the beginning of the calculation. The same procedure is then repeated between the second and third row, followed by the third and fourth row of the matrix element in order to determine the second and third slack variable i.e., V_2 in between BS_2 and CS_2 and V_3 in between CS_2 and DS_2, as follows;

The formulae adopted to perform the calculation for the two slack variables are as given below.

$V_2 = d_2 + w_2$ where $d_2 = CS_1 - (BS_2 + w_1)$, $w_2 = BS_3 - (CS_2 + d_2)$
$V_3 = d_3 + w_3$ where $d_3 = DS_1 - (CS_2 + w_2)$, $w_3 = CS_3 - (DS_2 + d_3)$

However, should the value of either or both d and w for the two respective stages appear to be negative, a zero value is taken instead.

Step 4: The makespan for the batch process is calculated using the formula;

$\text{Makespan} = AS_1+AS_2+BS_2+CS_2+DS_2+DS_3+V_1+V_2+V_3$

The generalized mathematical expression for the matrix approach: From the procedure developed above, generalized mathematical expressions could be developed for the matrix approach particularly for the calculation of the slack variables and the makespan. The developed mathematical equations are applicable to n number of products and are shown below;

$$d_k = M_{i+1,1} - (M_{i,2} + w_{k-1}) \qquad i = (1,\dots n-1) \qquad \text{(i)}$$
$$w_k = M_{i,3} - (M_{i+1,2} + d_k) \qquad k = (1,\dots n-1) \qquad \text{(ii)}$$
$$V_k = d_k + w_k \qquad \text{(iii)}$$

Note that $M_{i,j}$ represent the value of the matrix element situated in row i and column j starting from 1,1. The respective values for the idle time, d and the waiting time, w calculated from equation (i) and (ii), are substituted into equation (iii) in order to calculate the value of the respective slack variable, V. The makespan is then determined using equation (iv) below.

$$\text{Makespan} = M_{1,1} + \sum_{i=1}^{n} M_{i,2} + M_{n,3} + \sum_{k=1}^{n-1} V_k \qquad \text{(iv)}$$

Application of matrix approach: The effectiveness of the application of the suggested matrix approach for scheduling flowshop process is illustrated with an example problem. The example problem has been taken from Ku and Karimi (1992) for which, they have developed an MILP formulation to solve for the optimum production sequence offering minimum makespan (Edgar *et al.*, 2001). Using the matrix formulation, the makespan of the optimum batch sequence as suggested by Ku and Karimi (1992) is recalculated.

Example 4: Makespan of four products using Matrix approach

	1	2	3
1	3.5	4.3	8.7
		V_1 (d_1+w_1)	
2	4.0	5.5	3.5
		V_2 (d_2+w_2)	
3	3.5	7.5	6.0
		V_3 (d_3+w_3)	
4	12.0	3.5	8.0

Using the matrix formulation, the slack variables V_1, V_2 and V_3 are determined using equations (i), (ii) and (iii).

$V_1 = d_1 + w_1 = 3.2$

where, $d_1 = 4.0 - (4.3+0) = -0.3 = 0$, $w_1 = 8.7 - (5.5+0) = 3.2$

$V_2 = d_2+w_2 = 0$

where, $d_2 = 3.5 - (5.5+3.2) = -5.2 = 0$, $w_2 = 3.5 - (7.5+0) = -4 = 0$

$V_3 = d_3+ w_3 = 4.5$

where, $d_3 = 12.0\,(7.5+0) = 4.5$, $w_3 = 6.0 - (3.5+4.5) = -2 = 0$

Subsequently, the makespan for the batch production sequence is calculated using Eq. iv.

Makespan = 3.5+4.3+5.5+7.5+3.5+8.0+3.2+0+4.5 = 40 h

The makespan is found to be the same as those obtained by Ku and Karimi (1992) which demonstrate the validity of the matrix approach. Again it should be noted that the makespan calculated is based on the production sequence producing the products in the order of product A, followed by product B, then product C and lastly product D.

Determination of optimum batch production sequence: In optimizing the batch scheduling for a given batch process, it is important that the optimum sequence or few

Table 4: Makespan of all possible sequences of products A, B, C and D in three stages

Production sequence	Makespan (h)	Production sequence	Makespan (h)	Production sequence	Makespan (h)
ABCD	40.0	BDCA	42.2	CBDA	43.2
ABDC	37.3	BDAC	42.2	CBAD	40.5
ADBC	40.5	BADC	39.0	DBAC	42.5
ADCB	36.5	BACD	37.3	DBCA	45.7
ACDB	34.8	CADB	40.5	DCBA	42.5
ACBD	40.0	CABD	38.0	DCAB	41.7
BCAD	40.5	CDBA	39.2	DACB	41.7
BCDA	41.7	CDAB	40.0	DABC	45.7

best sequences are firstly determined from all possible batch production sequences that could be derived. The number of possible batch production sequences could be determined using the following permutation rule:

$P(n) = n!$ where:
$P(n)$ = No. of possible batch production sequences
n = No. of products

For example, the number of possible batch production sequences for producing four products i.e., A, B, C and D is $P(4) = 4! = 24$. Based on the permutation result, the various possible sequences are then developed and these are shown in Table 4. Given the matrix approach ability to quickly calculate the makespan for a specified production sequence, the task of calculating the makespan for all the possible sequences could be done easily by repeating the procedure. Table 4 shows the makespan calculated using matrix approach for all the possible production sequences producing the specified four products with batch process recipes as stated in example 4. It is obvious from Table 4 that the production sequence ACDB has the lowest makespan i.e., 34.8 h, which is the same as determined by Ku and Karimi (1992) using MILP formulation. This proves that the developed matrix approach is able to calculate the makespan for a specified batch process with NIS transfer policy. The makespan calculation in Table 4 has been performed using matrix approach which was programmed on a Microsoft Visual C++ (Version 6.0)™.

CONCLUSIONS

The design of batch process scheduling appears to be rather complex in view of the various parameters involved during optimization. Often intermediate products are allowed to stay within the current process unit while waiting for the availability of the next process unit. This creates another dimension to the design problem of the batch process scheduling. Mathematical programming methods based on MILP and MINLP were used widely in the past in view of its ability to overcome the complex optimization problem while designing batch process scheduling. The matrix approach introduced in the present work uses simple formulation derived from the logic obtained through the use of Gantt chart for determining makespan. Computer programming helps to execute the required formulation swiftly allowing the makespan calculation to be done for the various potential solutions and at the same time, screened for the best few. Given the options and having considered all the constraints (including subjective ones), the designer could then rationalize the solutions and select the best. Future work is planned for developing the approach further to account for other transfer policies such as the allocation of intermediate storage between process units. It is also envisaged that the matrix approach could be extended to incorporate uncertainties in product demand and supply.

REFERENCES

Balasubramanian, J. and I.E. Grossmann, 2002. A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing times. Comput. Chem. Eng., 26 (1): 41-57.

Biegler, L.T., I.E. Grossmann and A.W. Westberg, 1997. Systematic Methods of Chemical Process Design. Prentice-Hall.

Birewar, D.B. and I.E. Grossmann, 1989. Efficient optimization algorithms for zero-wait scheduling of multiproduct batch plants. Ind. Eng. Chem. Res., 28 (9): 1333-1345.

Das, H., P.T. Cummings and M.D. Le Van, 1990. Scheduling of serial multiproduct batch processes via simulated annealing. Comput. Chem. Eng., 14 (12): 1351-1362.

Edgar, T.F., D.M. Himmelblau and L.S. Lasdon, 2001. Optimization of Chemical Processes. 2nd Edn. McGraw Hill.

Grau, R., A. Espuna and L. Puigjaner, 1996. Completion times in multipurpose batch plants with set-up, transfer and clean-up times. Comput. Chem. Eng., 20 (2): S1143-S1148.

Jung, J.H., H.K. Lee, D.R. Yang and I.B. Lee, 1994. Completion times and optimal scheduling for serial multi-product processes with transfer and set-up times in zero-wait policy. Comput. Chem. Eng., 18 (6): 537-543.

Kim, M., J.H. Jung and I.B. Lee, 1996. Optimal scheduling of multiproduct batch processes for various intermediate storage policies. Ind. Eng. Chem. Res., 35 (11): 4058-4066.

Ku, H.M. and I.A. Karimi, 1988. Scheduling in serial multiproduct batch processes with finite interstage storage: A mixed integer linear program formulation. Ind. Eng. Chem. Res., 27 (10): 1840-1848.

Ku, H.M. and I.A. Karimi, 1990. Completion time algorithms for serial multiproduct batch processes with shared storage. Comput. Chem. Eng., 14 (1): 49-69.

Ku, H.M. and I.A. Karimi, 1992. Multiproduct Batch Plant Scheduling. In: CACHE Process Design Case Studies, Grossmann, I. and M. Morari (Eds.). CACHE Corporation, Austin, TX.

Orçun, S., I.K. Altinel and Ö. Hortaçsu, 2001. General continuous time models for production planning and scheduling of batch processing plants: Mixed integer linear program formulations and computational Issues. Comput. Chem. Eng., 25 (2-3): 371-389.

Ryu, J.H. and E.N. Pistikopoulos, 2007. A novel approach to scheduling of zero-wait batch processes under processing time variations. Comput. Chem. Eng., 31 (3): 101-106.

Suhami, I. and R.S.H. Mah, 1981. An implicit enumeration scheme for the flowshop problem with no intermediate storage. Comput. Chem. Eng., 5 (2): 83-91.

Voudouris, V.T. and I.E. Grossmann, 1992. Mixed-integer linear programming reformulations for batch process design with discrete equipment sizes. Ind. Eng. Chem. Res., 31 (5): 1315-1325.