Application of Heuristics to Solve Scheduling Problems of Intermediate Transfer Policies in Multiproduct Chemical Batch Processes

A. Shafeeq, A. Muhammad, R.U.Khan, M. Azam
Institute of Chemical Engineering and Technology,
University of the Punjab, Lahore, Pakistan.

Abstract

Batch processes are considered to be very efficient in producing fine as well as specialty chemicals. The efficiency of batch processes is attributed to the scheduling of various tasks involved in the production of a desired product. A very common purpose of scheduling is to reduce the total completion time of the process and is referred to as makespan. One of the ways to reduce makespan is the selection of a proper production sequence i.e. a sequence in which the raw materials are processed to produce specific products. The determination of such production sequence becomes a time consuming task with increase in the number of products. The complexity further increases when dealing with various transfer policies used for the transfer of product intermediates during the production cycle. Although numerous techniques are available but most of them are based on complex mathematical equations and thus take longer CPU time to solve even for a small batch scheduling problem. Further, the search of the optimal solution is not an easy task when the number of optimal solutions increases with increase in problem size. The motivation behind current work is to reduce the mathematical complexity as well as suggesting some rule based guidelines that could speed up the solution procedure for any batch scheduling problem. A new heuristic approach is developed and applied to various problem sizes in conjunction with mathematical formulations developed in our previous work for various transfer policies. The results so obtained are very promising and shows significant contribution towards the solution of batch scheduling problems with less computational effort.

***Keywords**: Batch process, Scheduling, Heuristics, Transfer Policy*

Introduction

Batch processes are usually preferred in process industry where production volume is low particularly for the production of paint, food, pharmaceuticals and specialty chemicals. Batch processes could be multiproduct, where all the products follow the same sequence of operation or multipurpose, where products need not to follow same operation sequence. Further the selection of inter stage transfer policy to transfer product intermediates from one stage to another is also very important in scheduling decisions. The usually referred transfer policies are zero wait (ZW) where the nature of the intermediate product demands its immediate transfer to the next stage. In contrast to ZW, NIS (no intermediate storage) transfer policy offers more flexible operation where product intermediates can wait inside the same stage until the next stage becomes available In addition to ZW and NIS transfer policies, other transfer policies considered are based on using the intermediate storage tanks in between the process units. The purpose of using storage tanks is to increase the plant availability by reducing the idle time of process units. The location as well as number of storages used depends on the type of products being produced and also on the economics of production. Further, the risks of storing the products inside the storage tanks for unnecessary time may result in changing the physical properties of the stored product with respect to time. Therefore, attention must be given to the storable time of the product intermediates while making scheduling decisions. In this context, the transfer policies adopted are usually referred to as UIS (unlimited intermediate storage) and FIS (finite intermediate storage). In UIS, there is no limit on the number of storage tanks i.e. storage tank is always available at the time of need. Whereas in FIS, the number of storage tanks is limited and are shared in case needed to store more than one product intermediate at one time (Grossmann, 1992; Kim et al., 1996; Moon et al., 1996; Ryu et al., 2007).

One of the important parameter that needs to be specified in scheduling operations is the selection of production sequence. The production sequence controls the completion time of the process which is also known as makespan in the published literature. The makespan varies when sequence of products to be produced in a batch facility is changed. The best sequence is the one that gives least makespan. For this purpose, the makespan for all the possible production sequences has to be determined first before the production sequence with minimum makespan is found. The calculation procedure becomes tedious with increase in number of products for different intermediate transfer policies discussed earlier. A number of scheduling techniques have been proposed. These include mathematical as well as heuristics. However, all the available techniques do not always ensure the global optimal solution and in many cases, produce near optimal solutions (Ku and Karimi, 1991; Balasubramanian and Grossmann, 2002).

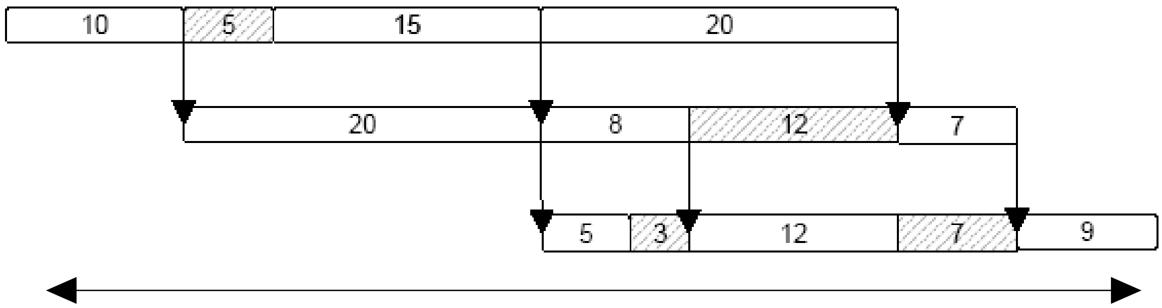
The heuristic rules developed in this work are found very promising in most of the example batch process recipe and also able to produce optimal solution with less computational effort. With the help of these heuristic rules, only partial enumeration is required i.e. numbers of possible production sequences searched for optimality are less than those of complete enumeration. The optimal production sequence in the present study is the one that produces minimum value of makespan. The makespan for any production sequence with various intermediate transfer policies could be determined using various mathematical formulations available in literature. This work uses the mathematical formulations developed in our earlier work (Shafeeq et al., 2008a,b) and summarized below for various intermediate transfer policies. The work presented here is a valuable extension to our previous work (Shafeeq et al.,2008c) and presents some more examples using different batch process recipes with various problem sizes. Further, a flowchart is also presented at the end for betterunderstanding.

Mathematical Formulations

ZW Transfer Policy

This policy requires the product intermediates to be transferred from one stage to the next as soon as they are produced as shown in Figure 1. This procedure could produce idle time between process stages as shown by shaded area in Figure 1. The idle time represents the time during which the stage remains idle or not in use. The determination of these idle times can be done using equation 1-3 below. The variable M and V with respective subscript numbers represent the stage and idle time location respectively. The makespan can be determined using equation 4. The idle time between stages has been shown using shaded area in Figure 1.

Figure 1. Gantt chart for three products in three stages for ZW Transfer Policy



$$V_{i,2} = M_{i+1,1} - M_{i,2} \text{ ....(1)} \qquad i = 1......n-1$$

$$V_{i,j+1} = (V_{i,j} + M_{i+1,j}) - M_{i,j+1} \text{.......(2)} \qquad j = 2........m-1, \quad i = 1......n-1$$

$$V_{i,j} = (V_{i,j+1} - M_{i+1,j}) + M_{i,j+1} \text{...........(3)} \qquad j = m-1........1, \quad i = 1......n-1$$

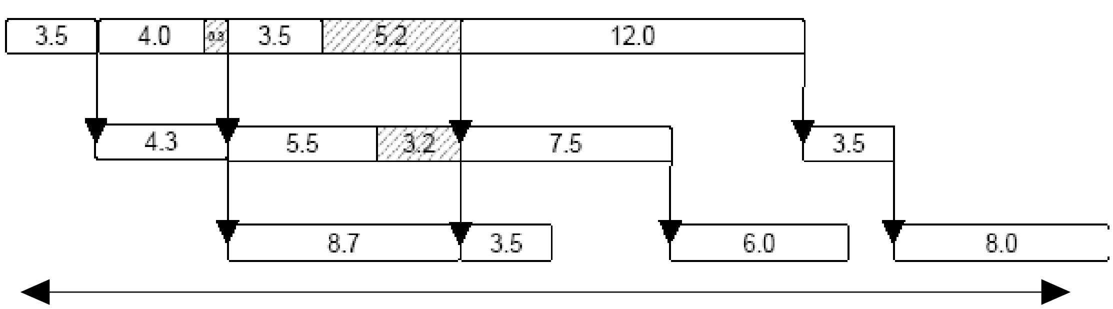
$$Makespan = \sum_{j=1}^{m} M_{1,j} + \sum_{i=2}^{n} M_{i,m} + \sum_{i=1}^{n-1} V_{i,m} \text{..................(4)} \qquad n \geq 2 \text{ , } m \geq 2$$

NIS Transfer Policy

In this transfer policy, the flexibility is provided in terms of allowing the product intermediate to wait inside the same stage till the next stage is available to accept the product intermediate from the previous stage. Figure 2 represents the waiting time of an example batch process recipe shown by the shaded area. The idle time (V) between the process stages and holding time (I) inside the same stages (shown by shaded area in Figure 2) can be calculated using the equations 5-6. The makespan of such batch process recipe with NIS transfer policy can be determined using the same equation used earlier for ZW transfer policy.

Makespan =

Figure 2 Gantt chart for four products in three stages for NIS Transfer Policy



$$V_{1,j+1} = (V_{1,j} + M_{2,j}) - M_{1,j+1}........................(5) \qquad j = 1........m-1$$

$$V_{i,j+1} = (V_{i,j} + M_{i+1,j}) - (M_{i,j+1} + I_{i-1,j+1})........(6) \qquad j = 1........m-1, \quad i = 2.......n\text{-}1$$

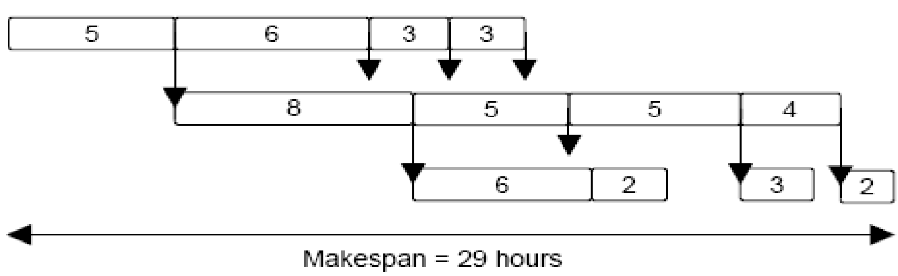
$$V_{i,1} = 0, \quad I_{i,m} = 0 \qquad i = 1.......n\text{-}1$$

If any of the value of $V_{i,j+1}$ $(j = 1........m-1,\ i = 1.......n\text{-}1)$ is found to be negative, the slackvaraible "V" is adjusted to zero between the particular stages and the value is assigned to $I_{i,j}$ $(j = 1........m-1,\ i = 1.......n\text{-}1)$ i.e. holding time in the preceding stage,otherwise holding time will be zero.

UIS Transfer Policy

Sometimes the nature of the product intermediates is such that they can not be held inside the same stage till next stage is made available. In such a case, a storage tank is used to temporary store the product intermediate till the next stage becomes available. The number of storage tanks is not limited and always available whenever required. The number of storage tanks and waiting time (W) inside the storage tanks (shown by inverted arrows in Figure 3) could be determined using equations 7-8. The makespan is determined using the same equation used earlier for the case of ZW and NIS transfer policy.

Figure 3. Gantt chart for four products in three stages for UIS Transfer Policy



$$V_{1,j} = (M_{2,j-1} + V_{1,j-1}) - M_{1,j}.................(7) \qquad j = 2......m$$

$$V_{i,1} = 0 \qquad i = 1......n-1$$

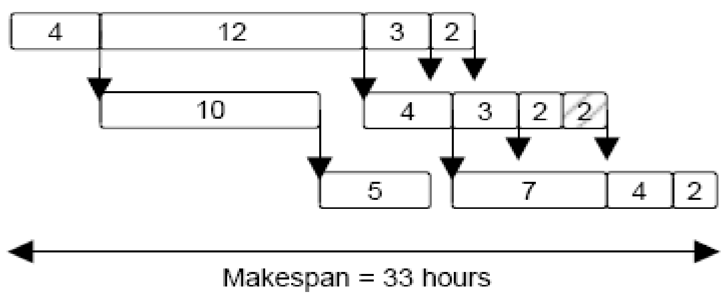
$$V_{i,j} = \left(\sum_{k=2}^{i+1} M_{k,j-1} + \sum_{k=1}^{i} V_{k,j-1}\right) - \left(\sum_{k=1}^{i} M_{k,j} + \sum_{k=1}^{i-1} V_{k,j}\right)........(8) \qquad j = 2.......m, \quad i = 2........n-1$$

If any value of $V_{1,j}$ $(j = 2....m)$ and $V_{i,j}$ $(j = 2.......m,\ \ i = 2........n-1)$ is negative, the value of zero is assigned to the slack variable "V" between the particular stages and that value will be assigned to $W_{1,j-1}$ $(j = 2....m)$ and $W_{i,j-1}(j = 2.......m,\ \ i = 2........n-1)$ respectively.
This represents waiting time inside the temporary storage for the preceding stage;
no temporary storage is needed after the preceding stage and "W" will be zero.

FIS Transfer Policy

This transfer policy is same as that of UIS transfer policy in a sense that it provides the facility of temporary storages to the product intermediates. However, the number of storages tanks is limited and not necessarily available all the time. In case, storage tank is not available, the product intermediates must be held inside the same stage till the next stage becomes available or storage tank is free as shown by shaded area in Figure 4 below. The number of storage tanks, holding time inside the same stage (I) and waiting time (W) inside the storage tanks (shown by shaded area and inverted arrows in Figure 4 respectively) can be determined using equations 9-10. Again, the makespan is determined using equation 4 as shown earlier for ZW, NIS and UIS transfer policies.

Figure 4. Gantt chart for four products in three stages for FIS Transfer Policy



$$V_{1,j} = (M_{2,j-1} + V_{1,j-1}) - M_{1,j} \text{..........(9)} \qquad j = 2........m$$

$$V_{i,1} = 0,\ \ I_{1,j} = 0 \qquad i = 1..........n\text{-}1,\ \ j = 1.........m$$

$$I_{i,m} = 0 \qquad i = 2..........n\text{-}1$$

$$\left[\begin{array}{l} if\ \ (M_{i+1,j-1} + V_{i,j-1}) \le W_{i-1,j-1},\ I_{i,j-1} = W_{i-1,j-1} - (M_{i+1,j-1} + V_{i,j-1})\ \ else\ I_{i,j-1} = 0 \\ V_{i,j} = (\sum_{k=2}^{i+1} M_{k,j-1} + \sum_{k=1}^{i} V_{k,j-1} + \sum_{k=1}^{i} I_{k,j-1}) - (\sum_{k=1}^{i} M_{k,j} + \sum_{k=1}^{i-1} V_{k,j} + \sum_{k=1}^{i-1} I_{k,j}) \\ \qquad\qquad\qquad\qquad\qquad\qquad j = 2......m,\ i = 2......n\text{-}1 \end{array}\right] \text{............(10)}$$

if any value of $V_{1,j}$ $(j = 2.....m)$ and $V_{i,j}$ $(j = 2....m,\ i = 2....n-1)$ is negative, the value of zero is assigned to the slack variable “V” between the particular stages and that value will be assigned to $W_{1,j-1}$ $(j = 2....m)$ and $W_{i,j-1}$ $(j = 2....m,\ i = 2....n-1)$ respectively i.e. waiting time inside the intermediate storage tank otherwise no intermediate storage tank is needed and "W" will be zero. Finally, the makespan will be calculated using equation (3).

Heuristic Approach

The development of heuristic rules has been shown for zero wait (ZW) transfer policy. The same rules would be applicable for batch processes that follow transfer policies discussed earlier i.e. NIS, UIS and FIS.

Example

This example shows the makespan calculation for a batch process producing three products namely A, B and C. The makespan for a batch process could be determined using equations 1-4 described earlier in the text. The batch process recipe is shown in Table 1. The value of makespan calculated for all possible sequences for products A, B and C are shown in Table 2. It could be observed that optimal sequence is BAC with minimum value of makespan i.e. 61 hours. The above procedure would become computationally expensive with increase in the number of products. Therefore, a heuristic procedure is developed in the present work to limit the number of sequences needed to be evaluated to find the sequence with minimum makespan.

TABLE 1

Processing time of three products in three stages for production sequence ABC

Products	Processing time (hr)		
	S_1	S_2	S_3
A	10	20	5
B	15	8	12
C	20	7	9

TABLE 2

Makespan for all possible sequences of A, B & C

Production Sequence	Makespan (hr)
ABC	66
ACB	65
BAC	61
BCA	70
CAB	70
CBA	70

Development of the Heuristic Rules

Two observations are made from the above examples that could be used as a basis for the heuristic rules developed in our earlier work (Shafeeq et al., 2008c).

1. The optimal sequence can start with the product that has the least makespan in the first stage.

2. The optimal sequence can start with the product that has the sum of its processing recipe and processing time in the last stages of all other products with the least value compared to the value when calculated for other products using the same procedure.

The following examples would illustrate the application of these rule to identify the possible optimal solutions.

	S_1	S_2	S_3
A	10	20	5
B	15	8	12

C	20	7	9

1. Product A has the least processing time for its first stage i.e. 10, compared to other products.
2. Determine the sum of processing recipe of the first product and processing time in the last stages of all other products. Repeat the same procedure for second and third product as shown below.

Consider product A is placed first.

$AS_1+AS_2+\underline{AS_3+BS_3+CS_3}$ = 10+20+5+12+9 = 56

Consider product B is placed first.

$BS_1+BS_2+\underline{AS_3+BS_3+CS_3}$ = 15+8+5+12+9 = 49

Consider product C is placed first.

$CS_1+CS_2+\underline{AS_3+BS_3+CS_3}$ = 20+7+5+12+9 = 53

The observations made from of point 1 and 2 above reveals that optimal production sequence should have either product A or B as the first product in the sequence. Therefore, the number of production sequences required to be searched for optimality should be four i.e. each starting with product A and B. This can be validated from Table 2 where sequence BAC has the least makespan. A generalization of these rules has been developed in our earlier work (Shafeeq et al., 2008c) and is presented here for the purpose of reference.

Generalized heuristic rules

Step 1: $Min\left(P_{i,1}\right) \qquad (i = 1..........n)$

Find the product that has the least processing time in the first stage

Step 2: $P_i = \sum_{j=1}^{m-1} S_{i,j} + \sum_{k=1}^{n} S_{k,m}$

Detemine the sum of processing recipe of the first product and the processing time in thelast stages of all other products. Repeat the same procedure with second and third product

Step 3: $Min\left(P_i\right)$

Find the minimum value from the values calculated in step 2 and the corresponding product

$Condition\ 1. \quad if\ \ 'i'\ in\ P_i\ and\ P_{i,1}\ is\ same$

If the same product is found in step 1 and 3, the partial enumeration would be carried out to produce only those sequences that place this product as the first product in the sequence

$$Number\ of\ enumeration\ \text{needed}: \ \frac{n!}{n}\times 1$$

$Condition\ 2. \quad if\ \ 'i'\ in\ P_i\ and\ P_{i,1}\ is\ different$

If the product(s) found in step 3 differs from step 1, the partial enumeration should have all the sequences begining with all the products identified in step 1 and 3 as the potential sequences to be screened.

$$Number\ of\ enumeration\ \text{needed}: \ \frac{n!}{n}\times (no.\ of\ times, the\ 'i'\ is\ different)$$

A number of examples are solved to demonstrate the effectiveness of the heuristic rules developed above. This has been done using a computer code developed for this purpose in Microsoft Visual C++™ on an Intel Pentium® IV CPU 2.40 GHz. The screen output of the developed computer code is shown in Figure 6 (a,b,c).

i) n = 4, m = 4

	S_1	S_2	S_3	S_4	SUM
P_1	14	45	49	37	239
P_2	36	11	37	44	215
P_3	29	35	50	30	245
P_4	45	30	19	20	225

Total enumeration possible = 24 CPU Time = 0.01 sec
Optimal production sequence obtained = $P_2P_1P_3P_4$ = 244 hours

Using Heuristics:

Minimum S_1 = 14, Minimum SUM = 215
Possible optimal production sequences = Starting with products P_1 or P_2
Enumeration recommended by heuristics= 12 i.e. 6 with each of P_1 and P_2
Optimal production sequence obtained = $P_2P_1P_3P_4$ = 244 hours
CPU Time = 0.005 sec
Reduction in solution search space = 50%

ii) n = 7, m = 4

	S_1	S_2	S_3	S_4	SUM
P_1	46	16	21	44	340
P_2	22	18	27	45	324
P_3	33	45	26	26	361
P_4	30	40	24	43	351

P_5	44	30	18	15	349
P_6	10	31	42	35	340
P_7	39	40	19	49	355

Total enumeration possible = 5040 CPU Time = 1.642 sec
Optimal production sequence obtained = $P_2P_1P_6P_4P_7P_3P_5$ = 335 hours

Using Heuristics:

Minimum S_1 = 10, Minimum SUM = 324
Possible optimal production sequences = Starting with products P_2 or P_6
Enumeration recommended by heuristics = 1440 i.e. 720 with each of P_2 and P_6
Optimal production sequence obtained = $P_2P_1P_6P_4P_7P_3P_5$ = 335 hours
CPU Time = 0.469 sec
Reduction in solution search space = 71%

iii) n = 8, m = 6

	S_1	S_2	S_3	S_4	S_5	S_6	SUM
P_1	21	24	44	26	19	14	325
P_2	18	11	32	31	11	17	294
P_3	38	18	20	25	26	25	318
P_4	34	12	24	47	41	12	349
P_5	11	22	38	30	26	14	318
P_6	17	26	47	27	45	49	353
P_7	45	49	13	29	34	18	361
P_8	25	36	11	28	14	42	305

Total enumeration possible = 40320 CPU Time = 21.017 sec
Optimal production sequence obtained = $P_5P_6P_4P_1P_7P_8P_3P_2$ = 417 hours

Using Heuristics:

Minimum S_1 = 11, Minimum SUM = 294
Possible optimal production sequences = Starting with products P_5 or P_2
Enumeration recommended by heuristics = 10080 i.e. 5040 with each of P_5 and P_2
Optimal production sequence obtained = $P_5P_6P_4P_1P_7P_8P_3P_2$ = 417 hours
CPU Time = 5.254 sec
Reduction in solution search space = 75%

iv) n = 9, m = 6

S_1	S_2	S_3	S_4	S_5	S_6	SUM

P_1	26	23	39	27	28	34	362
P_2	32	30	16	12	17	28	326
P_3	20	32	34	17	25	16	347
P_4	15	11	11	32	32	15	320
P_5	49	21	45	32	49	12	415
P_6	19	17	41	23	13	20	332
P_7	26	45	38	28	20	40	376
P_8	42	38	41	29	33	12	402
P_9	43	12	21	25	35	42	355

Total enumeration possible = 362880 CPU Time = 211.989 sec
Optimal production sequences obtained:
$P_4P_3P_9P_1P_5P_7P_8P_6P_2$, $P_4P_3P_9P_1P_7P_5P_8P_6P_2$,
$P_4P_6P_9P_1P_5P_7P_8P_3P_2$, $P_4P_6P_9P_1P_7P_5P_8P_3P_2$ = 449 hours

Using Heuristics:

Minimum S_1 = 15 Minimum SUM = 320
Possible optimal production sequences = Starting with products P_4
Enumeration recommended by heuristics = 40320 with each P_4
Optimal production sequences obtained:
$P_4P_3P_9P_1P_5P_7P_8P_6P_2$, $P_4P_3P_9P_1P_7P_5P_8P_6P_2$,
$P_4P_6P_9P_1P_5P_7P_8P_3P_2$, $P_4P_6P_9P_1P_7P_5P_8P_3P_2$ = 449 hours
CPU Time = 23.554 sec
Reduction in solution search space = 88%

v) n = 10, m = 7

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	SUM
P_1	22	45	11	17	46	27	35	434
P_2	38	29	32	28	17	37	25	447
P_3	20	49	13	50	35	33	20	466
P_4	22	45	43	44	50	43	25	513
P_5	35	23	45	29	10	33	24	441
P_6	30	14	16	21	44	49	19	440
P_7	20	15	15	47	39	15	14	417
P_8	32	49	33	21	34	12	38	447
P_9	40	40	46	45	39	36	46	512
P_{10}	13	37	29	36	46	13	20	440

Total enumeration possible = 3628800 CPU Time = 2090.657 sec
Optimal production sequence obtained = $P_6P_{10}P_5P_4P_9P_3P_8P_2P_1P_7$ = 580 hours

Using Heuristics:

Minimum S_1 = 13 Minimum SUM = 417
Possible optimal production sequences = Starting with products P_7 or P_{10}

Enumeration recommended by heuristics = 725760 i.e. 362880 with each of P_7 and P_{10}
Optimal production sequence obtained = $P_7P_{10}P_9P_4P_3P_8P_2P_6P_1P_5$ = 593 hours
CPU Time = 418.131 sec
Reduction in solution search space = 80%

Significant reduction in computational time could be observed for various problem sizes. However, the heuristic rules are not based on analytical approach. Therefore, there may be chances where the solution obtained using these heuristic rules for any specific batch process recipe could not be the one that meets the criteria of minimum makespan as observed in example batch process recipe (v) above. For this purpose, an extension to the heuristic rules has been suggested to consider more products for enumeration thereby increasing the possibility of finding the optimal solution. The selection of more products could be done by finding the products that correspond to the values next to the minimum values determined earlier in step 1 and 3 of heuristic rules. The makespan calculated is then compared with the previous makespan value. The same procedure could be repeated till the new makespan value obtained is greater than the previous one as shown in Figure 5. However, the need of further iteration would solely depend upon the process engineer's decision to find the optimal or near optimal solution.

Conclusion

The determination of optimal production sequence from a list of all possible sequences is not challenging if problem size is small. However for large problem sizes and when different transfer policies are concerned, the task of determination of optimal production sequences becomes tedious. The heuristic approach developed in this work limits the search for optimal solution by eliminating all those production sequence that would not likely to produce optimal solutions. This has been done by developing a set of heuristic rules. The results obtained for a number of example batch process recipes are shown to be promising and also able to reduce CPU time significantly.

Figure 5: Flowchart for partial enumeration

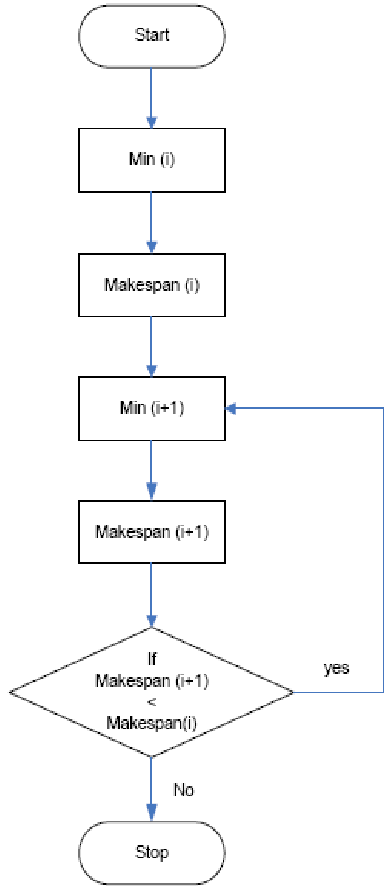


Figure 6a: Data input for the batch scheduling problem

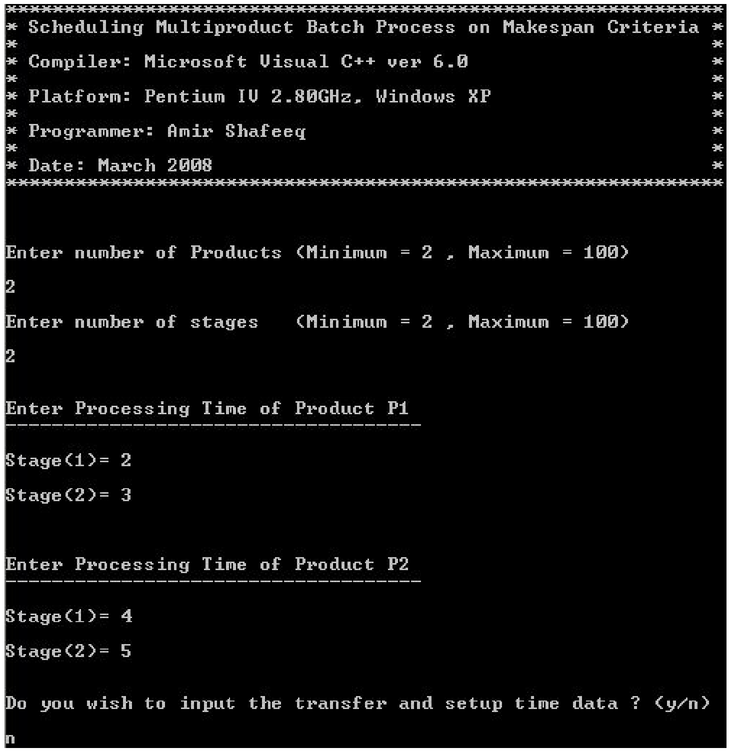
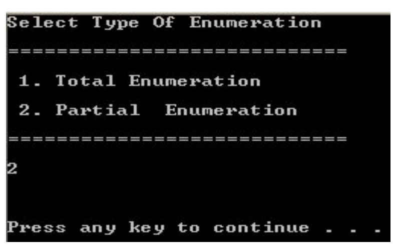


Figure 6b: Selection of transfer policy

Figure 6c: Selection of type of enumeration



References

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

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

Jung, J. H., Lee, H.K., Yang, D. R. and Lee, I.B. (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.

Ku, H. M. and Karimi, I.A. (1991). An evaluation of simulated annealing for batch process scheduling, Ind. Eng. Chem. Res., 30: (1) 163-169.

Lee, D. S., Vassiliadis, V. S. and Park, J. M. (2002). List-Based Threshold-Accepting algorithm for zero-wait scheduling of multiproduct batch plants, Ind. Eng. Chem. Res., 41: (25) 6579-6588.

Pitty,S.S. and Karimi,I.A.(2008).Novel MILP models for scheduling permutation flowshops,Chemical Product and Process Modeling,3:(1)1-46.

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

Shafeeq, A., Abdul Mutalib, M.I., Amminudin, K.A. and Muhammad A.(2008)a. New completion time algorithms for sequence based scheduling in multiproduct batch processes using matrix. Chemical Engineering Research and Design, 86,(10) 1167-1181.

Shafeeq, A., Abdul Mutalib, M.I., Amminudin, K.A. and Muhammad A. (2008)b. More on Completion Time Algorithms for Intermediate Storage Tanks in Multiproduct Batch Process Scheduling Using Matrix Representation. Ind. Eng. Chem. Res.,47, (24)9957–9970.

Shafeeq, A., Abdul Mutalib, M.I., Amminudin, K.A. and Muhammad A. (2008)c. “A heuristic method to search for optimal solution using partial enumeration for a multiproduct chemical batch process”. Proceedings of 22nd Symposium of Malaysian Chemical Engineers, Malaysia.