Available online at www.sciencedirect.com
ScienceDirect
Procedia MANUFACTURING
Procedia Manufacturing 2 (2015) 118 – 123

2nd International Materials, Industrial, and Manufacturing Engineering Conference, MIMEC2015, 4-6 February 2015, Bali Indonesia

Integer Batch Scheduling Problems for a Single-Machine to Minimize Total Actual Flow Time

Rinto Yusriski[a,b]* Budi Astuti[a], Sukoyo[b], TMA Ari Samadhi[b], Abdul Hakim Halim[b]
[a]Univesity of General Achmad Yani, Bandung, 40284 INDONESIA
[b]Bandung Institute of Technology, Bandung 40132, INDONESIA

Abstract

This research addresses a batch scheduling model for a single-machine under a Just-In-Time (JIT) production system that produces discrete parts. The objective is to minimize the total actual flow time, defined as the time when parts are flowing on the shop floor from its arrival time to their common delivery time. The decision variables are the number of batches, integer batch sizes, and the sequence of the resulting batches. The problem is solved based on the Lagrange Relaxation method. The optimality test of the proposed algorithm is done by comparing the result of the proposed algorithm with the Integer Composition method. The result of numerical experiments demonstrates that the proposed algorithm is very efficient to solve the problems.

Keywords: *batch scheduling, a single-machine, integer batch sizes, total actual flow time*

1. Introduction

In the manufacturing industry, several jobs of a similar family can be processed in a batch. There is a setup time between two consecutive batches. Processing jobs in batches is an important strategy to increase the capability of the machine and to decrease the total number of setups (see Baker and Trietsch [1]). For example, in the heat-treatment process, a number of jobs are processed in oven simultaneously in order to maximize the capacity of oven and minimize the consumption of energies. Another example, it can be seen in the Numerical Control (NC) machining process. Several parts of a similar family are proceeded in a batch so that the number of setups can be reduced.

Recently, batch scheduling problems have been discussed with several objectives, as in Dobson et al. [2], Dobson et al. [3], Halim et al [4], Ghazvini and Dupont [5], Lin and Jeng [6], Lin et al. [7], and Chung et al. [8]. The readers can refer to Potts and Kovalyov [9] to get an extensive review on scheduling with batching. Some of the authors, e.g., Dobson et al. [2], and Dobson et al. [3] have discussed batching problems on single-machine problems to minimize flow time. In Dobson et al. [2], flow time adopts the so-called the forward scheduling approach. The researchers assume that all parts have been available at time zero, and the parts in respective batches are delivered exactly at the completion time of the batches. On single-part single-machine scheduling problems, the trade-off is as follows. If all parts are processed in a single batch, the setup time can be reduced but the total flow time of parts will

Selection and Peer-review under responsibility of the Scientific Committee of MIMEC2015
doi:10.1016/j.promfg.2015.07.021

increase. However, if all parts are processed in the number of parts, the setup time will increase. In order to minimize the total flow time of parts, Dobson et al. [2] propose the formula for calculating batch sizes and the number of batches. The researchers show that the minimum flow time can be obtained by sequencing the resulting batches by using the Shortest Processing Time (SPT) rule.

The assumptions in Dobson et al. [2] are not always applied to real problems such as in a Just-In-Time (JIT) production system. There is a condition where the completed parts must be delivered exactly at the time coinciding with a due date, and the company is capable of managing the arrival of parts at the time when the machine starts processing. Halim et al. [4] propose an objective of actual flow time, defined as the time when parts are flowing on the shop floor from its arrival times to their common delivery time. The actual flow time adopts the backward scheduling approach. The application of the objective has been proved effectively to minimize total inventory cost in a Just-In-Time production system (see Halim and Ohta [10])

This research develops a model in Halim et al. [4] discussed a single-machine batch scheduling problems with non-integer batch sizes, this current research considers integer batch sizes. It is because, in the real system, there are situations where manufacturing systems produce discrete parts. The motivation of this research is the real-life situation in the production process of Enviplast. There is a step to print the Enviplast label by using printing machine. When the number of Enviplast reaches a certain discrete quantity, the operator must stop the machine and setup it again to refill the ink. Then, the quantity of Enviplast has been processed before the setup is called as one batch-size. However, the completed products in a batch are saved in a warehouse, and these are delivered to the consumer after the quantity reaches demand. Since all products must be delivered exactly at the common due date, this research uses the actual flow time as an objective.

The structure of this paper is as follows. The next section presents the problem formulation. The third section explores the solution method. The fourth section discusses several numerical experiments. Finally, the last is the remarks.

2. Problem Formulation

If there is a single-machine that processes *n* number of parts in *N* number of batches, thus there are integer batch sizes $Q_{[i]}$ where i=1,2,3, .., N. It is assumed that there is a setup time (s) between two consecutive batches. Further assumptions are: all parts in a batch are processed on the machine one by one consecutively with identical processing time (t), the arrivals of parts at the time when the machine start to process ($B_{[i]}$), and the delivery of finished parts at the common due date (d). If the objective is to minimize total actual flow time (F^a), Halim et al. [4] has been developed a mathematical model for non-integer batch sizes problems as follows.

$$\text{Minimizing } F^a = \sum_{i=1}^{N}\left(\sum_{j=1}^{i}\left(s + tQ_{[j]}\right) - s\right)Q_{[i]} \tag{1}$$

Subject to:

$$\sum_{i=1}^{N} Q_{[i]} = n, \tag{2}$$

$$\sum_{i=1}^{N} tQ_{[i]} + \left(N-1\right)s \le d, \tag{3}$$

$$B_{[1]} + tQ_{[1]} = d, \tag{4}$$

$$Q_{[i]}, N \ge 0, \quad i = 1,2,3,...,N \tag{5}$$

It is stated in Equation (1) that the objective is to minimize total actual flow time. Constraint (2) states that the number of all parts must be produced equal to the demand. Constraint (3) shows that all parts in the batches should be processed during the interval time from time zero to the due date. Constraint (4) states that the completion time of batch scheduled backwardly in the first order should be delivered precisely at the due date. Constraint (5) shows the non-negative constraint.

Halim, et. al [4] has been proposed the formula for calculating the number of batches and batch sizes and generate the optimal solution. For integer batch size problems, this research develops a model in Halim et al. [4] and propose new constraint as follows.

$$Q_{[i]} \geq 1 \text{ and integer} \tag{6}$$

$$1 \leq N \leq n \text{ and integer,} \tag{7}$$

Constraint (6) states the batch sizes should be larger than or equal to 1 and integer. Based on Equation (6), it is proposed Constraint (7), that is the number of batches should be a positive integer between 1 and demand.

3. Solution Method

This optimal solution can be found by generating all feasible batch size combinations by using the Integer Composition method (ICM) and then choosing the combinations with the minimum of total actual flow time. For example, if it is known the value of parameters as follows: t= 0.5, n= 3, d= 10, s= 1, the solution is as shown on Table 1 below.

Table 1: The Integer Composition Results

N	$Q_{[i]}$, i= (1, 2,...N)	$B_{[i]}$, i= (1, 2,...N)	F^a
1	(3)	(8.5)	4.5* (optimal solution)
2	(2,1)	(9, 7.5)	4.5* (optimal solution)
3	(1,2)	(8.5, 6.5)	8.5
4	(1,1,1,1)	(9.5, 8, 6.5, 5)	11

Table 1 shows that there are 4 integer batch-size combinations. The minimum total actual flow time is 4.5 when the numbers of batches are 1, and 2 with the batch size for N=1 is 3 and the batch sizes for N=2 are 2 and 1. Because the minimum of actual flow time is to be obtained from all feasible batch size combinations, the solution is optimal. However, it takes a long time to compute for generating all feasible batch size combinations. The total number of integer batch-size combinations can be counted by using the formula as follows.

$$f(n) = 2^{n-1} \tag{8}$$

Shen and Evan [11] have proved that the total number of integer batch-size combinations can be solved on a worst-case time complexity $O(n2^{n-1})$.

Preposition 1: batch scheduling problem for single-machine to minimize total actual flow time can be solved in a worst-case time complexity $O(n2^{n-1})$.

Proof: if there are n unit demands, the total number of integer batch-size combinations can be solved on $O(n2^{n-1})$. Because the total number of integer batch-size combinations can be counted using formula 2^{n-1}, the total number of batch-size transformations can be solved in $O(2^{n-1})$. If time complexity for searching the minimum of total actual flow time can be solved in $O(1)$. it can be solved in a worst-case time complexity $O(n2^{n-1})$. □ proven

Since the solution of batch size combinations is proved by Preposition 1 can be solved in a worst-case time complexity $O(n(2^{(n-1)})^2)$, this research proposes a heuristic algorithm based on the Lagrange Relaxation Methods and also proposes the formula for calculating the number of batches and batch sizes as follows.

$$N_{\max} = \min\left\{\left\lfloor \frac{d-nt}{s} \right\rfloor, \left\lfloor \frac{1}{2}+\sqrt{\frac{1}{4}+\frac{2nt}{s}} \right\rfloor + 1, n\right\} \tag{9}$$

$$Q_{[i]} = \max\left\{ \left\lceil \frac{n_{[i]}}{i} + \frac{1}{2}(i+1)\frac{s}{t} - \frac{s}{t}i \right\rceil, 1 \right\}, \quad i = N,...,1 \tag{10}$$

However, the minimum of total actual flow time can be obtained by sequencing the resulting batches by using the Longest Processing Time (LPT). It is proved by using Preposition 2 as follows.

Preposition 2: Minimizing total actual flow time in batch scheduling problems on single-machine can be obtained by sequencing the resulting batches by using the LPT rule in a backward scheduling approach.

Proof: There are two feasible schedules for N batches, and these are arranged by using backward scheduling approach. The first schedule places the batch of (i)-th position at the (i)-th position and the batch of (i+1)-th position to the (i+1)-th position. The last schedule is only different from the first schedule in the batch of (i)-th position at the (i+1)-th position and the batch of (i+1)-th position at the (i)-th position. If F^{a1} and F^{a2} are the values of total actual flow time for the first schedule and the second respectively, the following is obtained

$$F^{a1} - F^{a2} = \left(tQ_{[i]}Q_{[i+1]} + sQ_{[i+1]}\right) - \left(tQ_{[i+1]}Q_{[i]} + sQ_{[i]}\right)$$

Henceforth $\left(tQ_{[i]} + s\right)Q_{[i+1]} \leq \left(tQ_{[i+1]} + s\right)Q_{[i]}$

$$t_{[i]} + s / Q_{[i]} \leq t_{[i+1]} + s / Q_{[i+1]}$$

The conclusion: $\left(tQ_{[1]} + s\right)/Q_{[1]} \leq \left(tQ_{[2]} + s\right)/Q_{[2]} \leq ... \leq \left(tQ_{[N]} + s\right)/Q_{[N]}$

The value of total actual flow time on the left side will be minimum relative than that at the right side if a batch size on the left side is greater than that at the right side. Therefore, minimizing total actual flow time is obtained by sequencing the batches in order to use the LPT rule in the backward scheduling approach. □ Proven

Based on Eq. (8), (9) and Preposition 2, it is proposed a heuristic algorithm as follows.

Step 1. Input parameters n, d, t, s. Continue to Step 2.
Step 2. Find Nmax by Eq. (9). Determine j as an index for the number of batches (j = 1, …,Nmax). Continue to Step 3.
Step 3. Begin with j = 1. Determine $Q_{[1]} = n$. Continue to Step 4.
Step 4. Calculate total actual flow time $F_{[1]}$ by Eq. (1). Determine the value of $F_{[1]}$ as F^* (that is a temporary optimal solution). Continue to Step 5.
Step 5. Set j=j+1. If $1< j <N_{max}$, continue to Step 6, otherwise determine F^* as an optimal solution and then STOP. Continue to Step 6.
Step 6. Determine N=j. Continue to Step 7.
Step 7. Based on Preposition 2, calculate the batch sizes ($Q_{[i]}$) by using Eq. (10). Continue to Step 8.
Step 8. Compute the total actual flow time $F_{[j]}$ by Eq. (1). Continue to Step 9.
Step 9. Compare the value of total actual flow time $F_{[j]}$ and F^*. If $F_{[j]} < F^*$, determine $F_{[j]}$ as F^* and then return to Step 5, otherwise determine F^* as an optimal solution and then STOP.

4. Numerical Experiment

A numerical experiment test was conducted by giving the following input data parameters: t= 0.5, n = 20, d = 50, s = 2. The result of processing the proposed algorithm yields Nmax = 3. Table 2 presents the solution of the algorithm.

Table 2.Testing Results of The Heuristic Algorithm.

N	$Q_{[i]}$, i= (1, 2,...N)	$B_{[i]}$, i= (1, 2,...N)	F^a
1	(20)	(40)	200
2	(12, 8)	(44, 38)	168
3	(10, 7, 3)	(45, 39.5, 36)	165.5 * (The optimal solution)
4	(10, 7, 2,1)	(45, 39.5, 36.5, 34)	166.5

The search of solution start from the number of batches (N) that equals 1, and the result of total actual flow time is 200. The search is continued for N=2, and the result of total actual flow time is 168. The step is continued by comparing the value of the total actual flow time between N=2 and N=1. It is found that the total actual flow time for N=2 is smaller than N=1. Thus, the value of the total actual flow time for N=2 is defined as the temporarily optimal solution. The search of the solution is continued for N=3. The result of total actual flow time is 165.5. The step is continued by comparing the value of the total actual flow time between N=3 and N=2. It is found that the total actual flow time for N=3 is smaller than N=2. The value of the total actual flow time for N=3 is defined as the temporarily optimal solution. The search of the solution is continued for N=4. The result of total actual flow time is 166.5. The step is continued by comparing the value of the total actual flow time between N=4 and N=3. It is found that the total actual flow time for N=3 is smaller than N=4. The algorithm is stopped by determining the total actual flow time of N=3 as an optimal solution.

The tests are continued by testing the effectiveness of the heuristic algorithm by comparing the solution of the Integer Composition method (ICM) and the Heuristic Algorithm (HA). The Algorithms are coded in C++, and the results are reported by using the computer processor Intel Core i5-3337U, 1.8 GHz with 4 GB of Random Access Memory. The comparison tests have been done on more than 200 problems. If there are more than 300 unit demands, the solutions will not be found out because there is out of memory. In order to make well understanding about the behavior of comparison tests, Table 3 shows the examples of comparison tests for 10 problems with different input parameters as show as follows.

Table 3.Comparison Test between ICM and HA.

Input parameters				Total actual flow time		Time to Compute		Memory usage	
n	t	d	s	ICM	HA	ICM	HA	ICM	HA
6	10	1.4	0.9	28.0	28.0	< 1 second	< 1 second	1,4 KB	< 1 KB
8	13	1.2	0.6	33.0	33.0	< 1 second	< 1 second	2,91 KB	< 1 KB
9	15	1.3	0.7	47.1	47.1	< 1 second	< 1 second	8,44 KB	< 1 KB
10	18	1.5	0.8	65.7	65.8	00:00:01:0034	< 1 second	32,28 KB	1,3 KB
14	23	1.3	1.4	200.3	200.3	00:00:01:9931	< 1 second	102,8 KB	1,49 KB
15	26	1.7	1.2	201.2	201.2	00:00:02:0009	< 1 second	619,01KB	2,21 KB
16	26	1.3	0.8	154.1	154.1	00:00:03:0602	< 1 second	701,99 KB	2,66 KB
18	24	1.2	0.4	104.4	104.4	00:00:06:8201	< 1 second	1.92 MB	2,91 KB
21	29	0.8	0.9	281.6	281.6	00:00:08:3972	< 1 second	2.3 MB	3,20 KB
28	41	0.9	0.5	282.45	282.45	00:00:10:1996	00:00:01:0421	4.75 MB	4,42 KB

Table 3 shows that the Heuristic Algorithm produce the optimal solution because it has similar results with ICM algorithm. It can also be showed that for the need of time to compute and memory usage, the solutions of Heuristic Algorithm are more efficient than another one.

5. Concluding Remarks

The current research deals with a single-machine integer batch scheduling problems to minimize total actual flow time, and it can be solved in a worst-case time complexity $O(n2^{n-1})$. This paper proposes the heuristic algorithm that is developed by the Lagrange Relaxation method. The heuristic algorithm produces an optimal solution, and the application of the algorithm is very efficient to solve the problems. The extension of the research is under consideration and directed to multi-item case and multi-due date case.

Acknowledgements

The authors wish to thank the editors and the referees for their constructive comments and suggestions. This research was supported by the Bandung Institute of Technology (ITB) and the University of General Achmad Yani (UNJANI) Bandung with grant number SKEP/121/UNJANI/VI/2014.

References

[1] K. R. Baker, D. Trietsch, Principles of Sequencing and Scheduling, John Willey & Sons Inc, New Jersey, 2009.
[2] G. Dobson, U. S., Karmakar, and J. Rummel, Batching to minimize flow times on one machine, Management Science 33 (1987) 784-799.
[3] G. Dobson, U. S., Karmakar, and J. Rummel, Batching to minimize flow times on heterogeneous machines, Management Science 35 (1989) 607-613.
[4] A. H. Halim, S. Miyazaki, H. Ohta, Batch Scheduling Problem to Minimize Actual Flow Times of Component Through the Shop under JIT Environment, European Journal of Operational Research 72 (1994) 529-544.
[5] F. J. Ghazvini, L. Dupont, Minimizing Mean Flow Times Criteria on a Single Batch Processing Machine with Non-Identical Job Sizes, International Journal Production Economics 55 (1998) 273–280.
[6] B. M. T. Lin, A.A.K. Jeng, Parallel-Machine Batch Scheduling To Minimize the Maximum Lateness and the Number Tardy Jobs. International Journal of Production Economics 91 (2004) 121–134.
[7] B. M. T. Lin, T.C.E. Cheng, A.S.C. Chou, Scheduling in an Assembly Type Production Chain with Batch Transfer, Omega. 35 (2007) 143–151.
[8] S. H. Chung, Y. T. Tai, W. L. Pearn, Minimizing Makespan on Parallel Batch Processing Machines with Non-Identical Ready Time And Arbitrary Job Sizes, International Journal of Production Research 47 (2008) 5109–5128.
[9] C. N. Potts, M. Y. Kovalyov, Scheduling With Batching: A Review, European Journal of Operational Research 120, (2000) 228–249.
[10] A. H. Halim, H. Ohta, Batch Scheduling Problem to Minimize Inventory Cost in the Shop with Both Receiving and Delivery Just In Time, International Journal of Production Economics 33 (1994) 185-195.
[11] H. Shen, D. J. Evan, A New Method For Generating Integer Compositions In Parallel, Parallel Algorithms and Applications 9 (1996) 101-109.