A Flow Shop Batch Scheduling and Operator Assignment Model to Minimise Actual Flow Time

Dwi Kurniawan[1,2]*, Andi Cakravastia Radja[3], Suprayogi[4], Abdul Hakim Halim[5]
[1345] Graduate Program of Industrial Engineering
Bandung Institute of Technology, Bandung, Indonesia
Tel: (+62-22) 250 4189

[2] Study Program of Industrial Engineering
National Institute of Technology, Bandung, Indonesia
Tel: (+62-22) 727 2215

Email: [2]dwi_kurniawan@itenas.ac.id, [3]andi@mail.ti.itb.ac.id, [4]yogi@mail.ti.itb.ac.id, [5]ahakimhalim@mail.ti.itb.ac.id

Abstract. This paper discusses simultaneous problems of batch scheduling and operator assignment. There are a number of parts to be processed in batches where each batch is to be processed through a number of operations for which there are alternative operators with different set up and processing times. Each operator will be assigned at most to one machine. The decision variables are which operators that should be assigned to available machines, the number of batches, batch sizes and the schedule of the resulted batches. The proposed algorithm works by trying different number of batch, starts from one, until the best objective function value found. Numerical examples show that the model tends to assign the best operator to the machine with the longest processing time. Solutions obtained in this paper are local-optimal.

Keywords: Batch scheduling, operator assignment, flow shop, actual flow time.

1. INTRODUCTION

Scheduling is an activity of allocating a set of entities (tasks, events, vehicles or people) to a number of resources over time to achieve certain goals and to meet a set of constraints (Pinedo, 2002). Machine scheduling and operator scheduling problems have been considered as a complex problem (Pinedo, 2002), so both are frequently investigated separately. To improve industry's performance, scheduling research needs to consider machine and operator simultaneously (Van den Bergh et al., 2013).

Scheduling that considers machine and operator simultaneously has been investigated in several papers, such as Mencía et al. (2015) and Frihat et al. (2014). Mencía et al. (2015) propose a simultaneous scheduling model of machine and operator considering operations sequence to minimise makespan in job shop production systems, while Frihat et al. (2014) propose scheduling model to minimise personnel cost considering time lag between operations and due date. In both papers, processing times are assumed to be fixed.

In industry, processing time of an operation may vary. Kellerer and Strusevich (2008) and Grigoriev and Uetz (2005) propose scheduling model with a process that can be accelerated by allocating additional resources, such as tools. Process acceleration by allocating additional operator is proposed by Aftab et al. (2012) to minimise makespan, by Chaudhry (2010) to minimise flow time, and by Chaudhry and Drake (2009) to minimise total tardiness.

Variations in processing time may also come from different skill level between operators. This occurs for example in craft industries, where operator skill is essential for production, and there are alternative operators to perform these activities. To perform an operation in a machine, operator with better skill level will be able to do it in a shorter time. This situation has been investigated by Costa et al. (2013) who propose job scheduling model in single-stage system with different operator's set up times to minimise makespan. However, in craft industries, there are products that require processing in multi-stage systems with uniform routing (flow shops), and there are systems that perform production in batches. These situations have not been considered in Costa et al. (2013), thus it needs further development.

The literature shows that existing papers that consider machine and operator only discuss job scheduling as the object, while the current studies on batch scheduling only consider machine as the resource. Complexity may appear when a flow shop system performs production in batches, and there are alternative operators with different processing times for each operation. This situation requires batch

* Corresponding Author

scheduling and operator assignment to be performed simultaneously. This problem has not been investigated so far, and this research is intended to contribute in this area.

2. LITERATURE REVIEW

Based on the object, scheduling can be divided into job scheduling and batch scheduling. Job scheduling applies in scheduling where all parts are to be processed and moved in one group, while batch scheduling usually deals with dividing parts in job(s) into several groups, each to be processed and moved separately.

In batch scheduling that considers machine as the resource, Wang et al. (2016) discuss batch scheduling to minimise costs in single-stage system, while Liang and Hui (2016) propose models to minimise makespan, Ji et al. (2015) to minimise flow time, Halim and Ohta (1993) to minimise actual flow time, Li et al. (2015) to minimise number of tardy jobs, and Wang et al. (2016) to minimise total tardiness. For flow shops, Arroyo and Leung (2017) develop scheduling model to minimise makespan, while Bukchin et al. (2002) to minimise flow time, Halim and Ohta (1993) and (Halim et al., 1994) to minimise actual flow time. For job shops, the batch scheduling model is discussed by Mosheiov and Oron (2008) to minimise makespan and flow time. Some special situations have been considered in these models such as sequence-dependent set up in Liang and Hui (2016), deteriorating job in Ji et al. (2015) and energy cost restrictions in Wang et al. (2016).

Some papers have investigated job scheduling that considers machine and operator simultaneously. Costa et al. (2013) propose scheduling model in single-stage system to minimise makespan, Chaudhry et al. (2010) to minimise flow time, and Chaudhry and Drake (2009) to minimise total tardiness. Special situations are also considered in those papers, i.e. operators have multi-skill types in Costa et al. (2013), operators have multi-skill levels in Costa et al. (2013), and sequence-dependent set up in Costa et al. (2013), unfixed processing time due to process speed-up through additional operator in Zouba et al. (2009) and different skill levels between operators in Costa et al. (2013).

It can be seen from this review that existing papers considering machine and operator as resources still discuss job scheduling, while the current studies on batch scheduling only consider machine as the resource. This research will contribute a flow shop batch scheduling model that considers machine and operator simultaneously.

3. PROBLEM DEFINITION

The problem studied in this research is described as follows. There is a job consisting of *n* parts to be scheduled, it will be split into *N* batches, and each of batch *i* will be processed through *m* operations with uniform routing that can be performed by one of *o* alternative operators with different set up times $s_{k,w}$ and processing times $t_{k,w}$. The decision variables in the model are assignment of operator *w* to machine *k* ($X_{k,w}$), the number of batch *N*, batch sizes Q_i, and the schedule of operation *k* in batch *i* ($B_{k,i}$). All operations should be finished no later than due date *d*. This study will develop a model of simultaneous batch scheduling and operator assignment to minimise actual flow time in flow shop production systems.

Assumptions used in this study are:

1. All parts, machines and operators are ready (can be scheduled) at $t = 0$.
2. There is only one machine to perform each operation.
3. All operations have to be performed.
4. Due date for all parts required is known and fixed.
5. Set up is performed for each batch after the parts arrive.
6. Set up and processing times by each operator are known and fixed.
7. Interruption of an operation are not allowed.
8. Each machine and operator cannot perform more than one operation at a time.
9. Machines are always available during the scheduling time horizon.
10. Set up and process of a particular operation are performed by the same operator.
11. An operator is only assigned to maximum one machine.

4. MODEL FORMULATION

This research extends the flow shop batch scheduling model in Halim and Ohta (1993) by introducing unfixed set up and processing times depending on which operator is assigned to each machine. The existence of operator alternatives to perform operations is inspired by the single-stage job scheduling model proposed by Costa et al. (2013) which provides a simultaneous model of job scheduling and operator assignment. Variable and parameter notations, objective function and constraints formulation that related to batch scheduling are derived from Halim and Ohta (1993), while operator's alternative set up and processing times and operator assignment decision variables are adopted from Costa et al. (2013).

The problem defined in this research is modelled mathematically using a number of indices, parameters and variables as shown as follows.

Indices:
k = set of machines, $k = 1, ..., m$,
w = set operators, $w = 1, ..., o$,
i = batch sequence from the due date, $i = 1, …, N$.

Parameters:
n = number of parts to be processed (units),

m = number of machines (units),
o = number of operators (people),
d = due date, calculated from $t = 0$ (time),
$s_{k,w}$ = set up time per batch on machine k when performed by operator w (time),
$t_{k,w}$ = processing time per part on machine k when performed by operator w (time).

Variables:
F^a = total flow time of all parts (time-units),
N = number of batch,
$B_{k,i}$ = starting time of batch i processing in machine k (time),
Q_i = batch size, number of parts in batch i (units),
$X_{k,w}$ = binary variable that equals to 1 if operator k is assigned to machine w, equals to 0 if not,
W = set of operators assigned to machine 1 to k.

Batch scheduling and operator assignment problem to minimise actual flow time in flow shop production system can be formulated in model (1) to (11).

$$\min F^a = \sum_{i=1}^{N} \left(d - B_{1,i}\right) Q_i \qquad (1)$$

subject to

$$B_{m,i} = d - \sum_{w=1}^{o} X_{m,w} \left(i s_{m,w} + t_{m,w} \sum_{j=1}^{i} Q_j \right), \qquad i = 1, ..., N \qquad (2)$$

$$B_{k,1} = B_{k+1,1} - \sum_{w=1}^{o} X_{k,w} \left(s_{k,w} + t_{k,w} Q_1\right), \qquad k = 1, ..., m\text{-}1 \qquad (3)$$

$$B_{k,i} \le B_{k,i-1} - \sum_{w=1}^{o} X_{k,w} \left(s_{k,w} + t_{k,w} Q_i\right) \qquad k = 1, ..., m\text{-}1,\ i = 2, ..., N \qquad (4)$$

$$B_{k,i} \le B_{k+1,i} - \sum_{w=1}^{o} X_{k,w} \left(s_{k,w} + t_{k,w} Q_i\right), \qquad k = 1, ..., m\text{-}1,\ i = 2, ..., N \qquad (5)$$

$$B_{1,\mathrm{N}} > 0, \qquad (6)$$

$$\sum_{i=1}^{N} Q_i = n, \qquad (7)$$

$$\sum_{w=1}^{o} X_{k,w} = 1, \qquad k = 1, ..., m \qquad (8)$$

$$\sum_{k=1}^{m} X_{k,w} \le 1, \qquad w = 1, ..., o \qquad (9)$$

$$X_{k,w} = 0 \text{ or } 1, \qquad \forall k, w \qquad (10)$$

$$N \geq 1,\ Q_i > 0,\ i = 1, ..., N. \qquad (11)$$

Equation (1) is the objective function, minimisation of total actual flow time, the sum of multiplication between batch's actual flow time with the batch size. Batch's actual flow time is the time spent by a batch in the shop floor, from the batch arrives in the shop floor until its due date. Calculations of batch's starting time in each machine are stated in (2) to (5). Equation (2) states that batch's starting time in the last machine is the previous batch's starting time subtracted by the batch's set up and processing time. The first batch's process finishes at due date. Equation (3) states that the starting time of the first batch in the first machine until before-the-last machine is counted down from its starting time in the last machine. Constraints (4) and (5) state that the processing of batches other than the first batch in machines other than the last machine should be no later than the starting time of the previous batch in that machine and no later than the starting time of the batch in the next machine. It should be noted that set up and processing times in (2) to (5) are always multiplied by operator assignment variables, because there are alternative operators to perform each operation with different set up and processing times. Constraint (6) states that the starting time of the last batch in the first machine must be non-negative. Equation (7) states that the number of parts in all batches must equal to the total number of parts. Equation (8) states that each machine is operated by exactly one operator, while constraint (9) states that each operator can only be assigned to maximum one machine. Constraint (10) states that the operator assignment variables are binary numbers, and constraint (11) states that batch sizes must be positive and there must be a minimum one batch.

5. ALGORITHM

The problem formulated in (1) to (11) is not linear and not convex because it contains discrete variables $X_{k,w}$. Therefore, the problem will be solved by first setting the batch number, N. An algorithm is developed to solve the problem by trying several N values until the best objection function value is found. The solution method for the problem is described in the following algorithm.

Algorithm.
Step 1. Set $N = 1$. Continue to Step 2.
Step 2. Solve the problem in (1)-(11), determine Q_i and F^a, and set N, W and F^a as N^*, W^* and F^{a*}, the current best solution. Continue to Step 3.
Step 3. Set $N = N + 1$. Continue to Step 4.
Step 4. If $N = n$, continue to Step 6. If not, solve the problem in (1)-(11), determine N, Q_i and F^a, Continue to Step 5.
Step 5. If $F^a < F^{a*}$, set N, W and F^a as N^*, W^* and F^{a*}, back to Step 3. If not, continue to Step 6.
Step 6. Stop. The current F^{a*} is the optimal solution.

6. NUMERICAL EXAMPLES

The model and algorithm will be applied to 10 data sets, consisting of four 2x3 data sets (2 machines and 3 operators), two 3x4 data sets and four 4x4 data sets. Parameter of the ten data sets is written in Table 1. For all data sets, n values are set to be equal so the effect of other parameters on the results are more noteworthy.

Table 1: Parameter of data sets

Data set	n	m	o	d	$s_{k,w}$	$t_{k,w}$
1	100	2	3	1200	32 30 34 48 50 45	4 6 5 6 5 3
2	100	2	3	1200	32 30 34 48 44 45	5 4 6 6 7 5
3	100	2	3	1200	30 32 34 48 45 44	5 6 6 5 3 4
4	100	2	3	1200	30 32 34 48 45 15	5 6 5 5 3 3.5
5	100	3	4	1600	32 29 34 46 43 50 30 37 39 48 45 40	2 3 5 7 5 8 6 6 6 4 5 3
6	100	3	4	1600	32 26 34 46 43 50 30 31 39 38 45 40	2 2.3 5 7 6 8 5 5 6 4 5 3
7	100	4	4	3000	39 21 29 39 25 55 35 21 34 50 29 46 34 55 43 26	11 3 10 11 9 4 12 6 7 12 9 10 12 4 11 12
8	100	4	4	3000	87 41 52 81 72 59 89 82 59 84 29 42 53 92 3 55	14 3 10 12 2 12 8 6 9 8 6 10 3 11 4 2
9	100	4	4	3000	31 35 34 31 24 26 27 22 41 42 41 40 45 46 48 47	6.8 7 7 7.1 2 3 2 3 7 8 9 9 7 6 6 6
10	100	4	4	3000	54 56 59 52 41 43 44 39 64 67 68 63 71 72 75 69	3 4 5 2 6 7 8 5 9 10 11 8 12 13 14 11

Applying Algorithm to the data sets, the results are shown in Table 2. The algorithm is run by using the Lingo 12.0 software installed in a Core i5-6200U with 8 GB RAM computer.

As seen in Table 2, each data set achieves the best objective function at different N^* values. If the best solution is achieved at $N = N^*$, then the algorithm must be run from $N = 1$ to $N = N^*+1$, when the objective function value does not improve anymore. Therefore, computation time for each data set as shown in Table 2 is the sum of computation time from $N = 1$ until $N = N^*+1$.

Table 2: Computation results

Data set	m	N^*	W^*	$F^{a}*$	Computation time (days or h:m:s)
1	2	6	[1,3]	43503.2	0:03:54
2		8	[2,3]	53103.9	2.42 days
3		7	[1,2]	48061.6	0:49:48
4		6	[1,2]	43502.2	4:36:37
5	3	9	[1,3,4]	59163.7	11.75 days
6		9	[2,3,4]	53546.5	3.27 days
7	4	12	[3,4,1,2]	94094.6	4.04 days
8		9	[2,1,3,4]	71608.5	1.48 days
9		11	[4,3,1,2]	85350.6	14:06:07
10		9	[2,3,1,4]	132655.0	8:31:35

Table 3: Computation time for Data Set 7

N	$F^{a}*$	Computation time (days or h:m:s)
1	281900.0	0:00:00
2	167245.0	0:00:10
3	130224.1	0:00:11
4	113138.1	0:00:14
5	104040.4	0:00:15
6	98941.8	0:00:28
7	96279.8	0:01:21
8	94982.8	0:02:12
9	94382.8	0:09:56
10	94150.7	0:51:33
11	94094.8	4:00:59
12	94094.6	2.50 days
13	94094.6	1.33 days
Total		4.04 days

An example of implementing Algorithm is given for Data Set 7. We start Algorithm from Step 1 by setting $N = 1$, then continue to Step 2 by using this N to solve the problem formulated in (1)–(11), which result $W = [3,4,1,2]$ and $F^a = 281900$. We set $N = 1$, $W = [3,4,1,2]$ and $F^a = 281900$ as N^*, W^* and $F^{a}*$. Continuing to Step 3, we increase $N = 2$. Continuing to Step 4, because $N = 2$ have not reached $n = 100$, we solve problem (1)–(11) again by using $N = 2$, which result $W = [3,4,1,2]$ and $F^a = 167245$. In Step 5, we see that $F^a < F^{a}*$, so we set $N = 2$, $W = [3,4,1,2]$ and $F^a = 167245$ as N^*, W^* and $F^{a}*$, then go back to Step 3. Now we increase $N = 3$. We continue to Step 4, and because N have not reached n, we solve problem (1)–(11) again by using $N = 3$, and the result is $W = [3,4,1,2]$ and $F^a = 130224.1$. We see again in Step 5 that $F^a < F^{a}*$, so we set $N = 3$, $W = [3,4,1,2]$ and $F^a = 130224.1$ as the new N^*, W^* and $F^{a}*$, then go back to Step 3. Repeating Step 3 to 5 several times, we see that $F^{a}*$ continue to improve respectively to 113138.1, 104040.4, 98941.8, 96279.8, 94982.8, 94382.8, 94150.7, 94094.8, 94094.6, while $W = [3,4,1,2]$ does not change. At N = 12 and F^a = 94094.6, when we increase N = 13 and solve problem (1)–(11) again, we see that F^a does not improve anymore, so we stop, and the current N^* = 12, W^* = [3,4,1,2] and F^{a*} = 94094.6 become the optimal solution (underlined). The solution and computation time for Data Set 7 are shown in Table 3.

Solution that is resulted from Algorithm is not guaranteed to be a global optimal, it is only said as a local optimal. To convince that the solution is global optimal, all possible N values should be tried; the algorithm should stop when $N = n$. However, this will increase the computation time significantly. Thus, stopping at the first local optimum found is a trade-off to get a good solution in a reasonable computation time.

The final results of Algorithm implementation to data sets are the Gantt charts, consisting batch processing schedule in each machine. In Figure 1, Gantt chart for Data Set 7 is shown as an example. Operator assignment, batch numbers and starting times of each batch in each machine follow the result given by Lingo.

Table 4: Operator's rank for Data Set 7

		Total processing time				Operator's rank			
		w				*w*			
		1	2	3	4	1	2	3	4
k	1	1139	321	1029	1139	3	1	2	3
	2	925	455	1235	621	3	1	4	2
	3	734	1250	929	1046	1	4	2	3
	4	1234	455	1143	1226	4	1	2	3

From the computation results, it is notable that the best operator assignment (W^*) is obtained mostly by assigning the best operator to machine with the longest processing time (for ease, it is named as the BFL – best for longest – rule). The best operator for each machine is the operator whose the smallest total processing time, i.e. total time to set up and to process all parts without batching. For Data Set 7 as an example, each machine's best operators are shown by rank 1 in each row (underlined) in Table 4. The ranks (right table) are based on operator's total processing time which is given by operator's set up and processing time without batching (left table). Based on the ranks, operator 1 is assigned to machine 3 because (s)he is the best for it. Operator 2 is the best for machine 1, 2 and 4, but (s)he is assigned to machine 4 because it has the longest total processing time. For the remaining operators and machines, operator 3 is the best for machine 1 and operator 4 is the best for machine 2. So, operator assignment for Data Set 7 according to the BFL rule is W = [3,4,1,2], the same with W^* resulted by Algorithm. The values of W^* for the other data sets also assign the best operator to the longest total processing time machine, except in Data Set 6. In Data Set 6, operators' processing times differ slightly, so

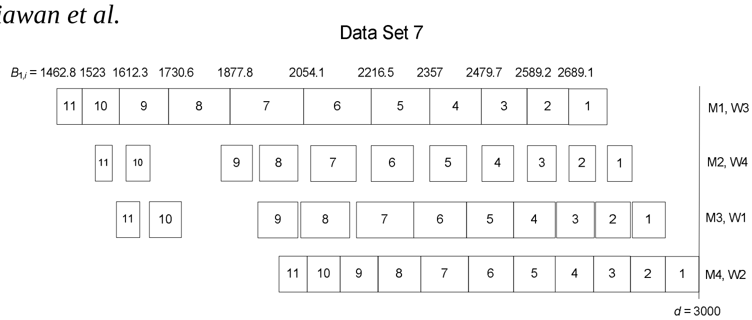

Figure 1: Gantt chart for Data Set 7

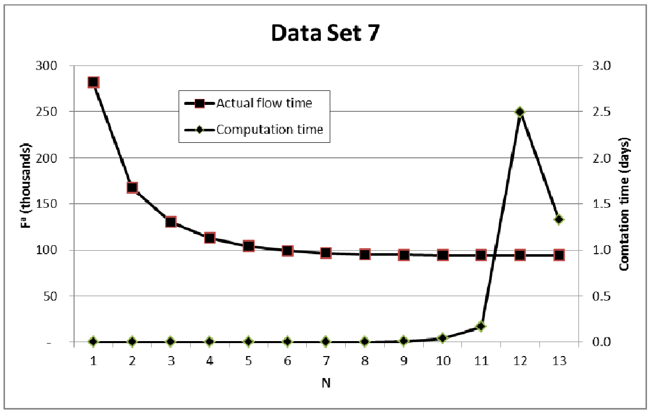

Figure 2: Actual flow time and computation time for Data Set 7

the best operator is only barely better than the others. This causes the BFL rule does not apply in Data Set 6.

It is also found that for each data set, the chosen operator assignment is the same for all N values. For example, in Data Set 7, according to Table 2 and 3, the best operator assignment is W = [3,4,1,2]. This assignment is the same for all tried N values, from N = 1 to N = 13. Thus, operator assignment is not affected by the number of batch.

Related to computation time, it is found that for each data set, computation time tends to increase when N increases, as seen in Table 3. This occurs because when N increases, the number of Q_i and $B_{k,i}$ variables involved in the model also increases, thus it takes more time to find the best solution. Comparison between actual flow time and computation time for different number of batch in Data Set 7 is shown in Figure 2. Additionally, from Table 2, it is shown that computation time for 5 of 10 data sets require more than one day. In practical implementation, these times are considered too long, so it is necessary to develop a new solution method that gives result quicker, which will be developed in a future research.

The model developed in this study can be applied for flow shop systems with any number of machines that have alternative operators in the process, and each operator can only be assigned to maximum one machine. The number of operators must be at least equal to the number of machines. In addition, this model can only be used if the set up and process of an operation is performed by the same operator.

7. CONCLUDING REMARKS

This study developed a batch scheduling and operator assignment model to minimise actual flow time in flow shop production systems. The model developed in this study can be used for all flow shop systems where there are alternative operators for each operation, and each operator can only be assigned to maximum one machine. The algorithm works by increasing the number of batch until the solution does not improve anymore. The model tends to assign the best operator to the machine with the longest processing time. The solution found using the algorithm is local optimal since not all possible numbers of batch are tried. For future research, it is necessary to develop a new solution method that can be performed quicker, so it could be more applicable in real industries. The proposed model should also be developed to accommodate assignment of operators to more than one machine. This will make the model more realistic and more applicable in industries.

REFERENCES

Aftab, M. T., Umer, M. & Ahmad, R. (2012). Jobs scheduling and worker assignment problem to minimize makespan using ant colony optimization metaheuristic. *Proceedings of World Academy of Science, Engineering and Technology***,** 1495.

Arroyo, J. E. C. & Leung, J. Y.-T. (2017). Scheduling unrelated parallel batch processing machines with non-identical job sizes and unequal ready times. *Computers & Operations Research,* 78**,** 117-128.

Bukchin, J., Tzur, M. & Jaffe, M. (2002). Lot splitting to minimize average flow-time in a two-machine flow-shop. *Iie Transactions,* 34**,** 953-970.

Chaudhry, I. A. (2010). Minimizing flow time for the worker assignment problem in identical parallel machine models using GA. *The International Journal of Advanced Manufacturing Technology,* 48**,** 747-760.

Chaudhry, I. A. & Drake, P. R. (2009). Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms. *The International Journal of Advanced Manufacturing Technology,* 42**,** 581-594.

Chaudhry, I. A., Mahmood, S. & Ahmad, R. (2010). Minimizing makespan for machine scheduling and worker assignment problem in identical parallel machine models using GA. *Proceedings of the World Congress on Engineering,* 3.

Costa, A., Cappadonna, F. & Fichera, S. (2013). A hybrid genetic algorithm for job sequencing and worker allocation in parallel unrelated machines with sequence-dependent setup times. *The International Journal of Advanced Manufacturing Technology,* 69**,** 2799-2817.

Frihat, M., Sadfi, C. & Hadj-Alouane, A. B. (2014). Optimization of integrated employee timetabling and Hybrid Job Shop Scheduling under time lag constraints. *International Conference on Control, Decision and Information Technologies (CoDIT)***,** 282-287.

Grigoriev, A. & Uetz, M. (2005). Scheduling parallel jobs with linear speedup. *International Workshop on Approximation and Online Algorithms***,** 203-215.

Halim, A. H., Miyazaki, S. & Ohta, H. (1994). Lot scheduling problems of multiple items in the shop with both receiving and delivery just in time. *Production planning & Control,* 5**,** 175-184.

Halim, A. H. & Ohta, H. (1993). Batch-scheduling problems through the flowshop with both receiving and delivery just in time. *The International Journal Of Production Research,* 31**,** 1943-1955.

Ji, M., Yang, Q., Yao, D. & Cheng, T. (2015). Single-machine batch scheduling of linear deteriorating jobs. *Theoretical Computer Science,* 580**,** 36-49.

Kellerer, H. & Strusevich, V. A. (2008). Scheduling parallel dedicated machines with the speeding-up resource. *Naval Research Logistics (NRL),* 55**,** 377-389.

Li, S., Cheng, T. E., Ng, C. & Yuan, J. (2015). Single-machine batch scheduling with job processing time compatibility. *Theoretical Computer Science,* 583**,** 57-66.

Liang, Y. & Hui, C. W. (2016). Simultaneous subtour elimination model for single-stage multiproduct parallel batch scheduling with sequence dependent changeovers. *Computers & Chemical Engineering,* 87**,** 63-67.

Mencía, R., Sierra, M. R., Mencía, C. & Varela, R. (2015). Genetic algorithms for the job shop scheduling problem with operators. *Applied Soft Computing,* 34**,** 94-105.

Mosheiov, G. & Oron, D. (2008). Open-shop batch scheduling with identical jobs. *European Journal of Operational Research,* 187**,** 1282-1292.

Pinedo, M. (2002). Scheduling: theory, algorithms, and systems. *Prentice Hall, USA.*

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E. & De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research,* 226**,** 367-385.

Wang, J., Qiao, F., Zhao, F. & Sutherland, J. W. (2016). Batch scheduling for minimal energy consumption and tardiness under uncertainties: A heat treatment application. *CIRP Annals-Manufacturing Technology.*

Zouba, M., Baptiste, P. & Rebaine, D. (2009). Scheduling identical parallel machines and operators within a period based changing mode. *Computers & Operations Research,* 36**,** 3231-3239.