

Modelling the Lot Streaming Problem with Setup times in a Job Shop Manufacturing System

Udo Buscher and Liji Shen

Abstract—This paper aims at solving the lot streaming problem in a job shop environment, where setup times are involved. The proposed integer programming formulation sufficiently describes the processing dynamics of individual sublots and enables the simultaneous determination of schedules on machines and subplot sizes. Small instances of job shop problems with consistent sublots can thus be optimally solved. Computational results confirm that, by applying the lot streaming strategy, both idling times of machines and completion times of operations are significantly reduced. In view of setups, various types of setups are incorporated in the model. The influence of setup times on the performance of lot streaming is also intensively examined. In addition, the efficiency of the formulation with special constraints is evaluated.

Index Terms—lot streaming; job shop scheduling; integer programming

I. INTRODUCTION

THE purpose of this paper is to solve the lot streaming problem in a job shop environment, where setup times are involved. The *job shop* scheduling problem can be briefly described as follows [11]: A set of jobs and a set of machines are given. Each machine can process at most one job at a time. Each job consists of a sequence of operations, which need to be processed during an uninterrupted time period of a given length on a given machine. A *schedule* is an allocation of the operations to time intervals on the machines. The objective is to find a schedule of minimum length (*makespan*). This class of problems has been proven to be NP-hard [12].

With respect to *lot streaming*, a job is actually a *lot* composed of identical items. In classical job shop scheduling problems a lot is usually indivisible. The entire lot must be completed before being transferred to its successor operation, which leads to low machine utilization and long completion times. Lot streaming techniques, on the other hand, provide the possibility of splitting a lot into multiple smaller sublots, which can be treated individually and immediately transferred to the next stage once they are completed. Different sublots of the same job can thus be simultaneously processed at different operation stages. As a result of operation overlapping, the production can be considerably accelerated. However, due to the complex interaction between sublots and machines, job shop problems with the application of the lot streaming strategy is difficult to formulate mathematically.

In the last years, a majority of researches focused on solving lot streaming problems in a flow shop production system. Heuristics for two machine flow shops with equal-sized sublots and a makespan objective function were first

discussed in [18]. Vickson and Alfredsson [21] considered two and three machine flow shop scheduling problems with unit-sized transfer lots (that is, each subplot contains only one unit). A modification of Johnson's Algorithm [16] for makespan minimization is proposed. Baker [1] extended the model established by [21] and derived optimal scheduling policies for the two-machine flow shop with lot streaming and setup times. Trietsch and Baker [20] presented an overview of basic models and solution algorithms for the lot streaming problem. Their study provided important insights into the solution structure for larger lot streaming problems.

For the m -machine production environment, Glass et. al. [13] used a network representation to analyse the structure of optimal solutions. Chen and Steiner [5] extended the results of [13] to further consider detached setup times. Chen and Steiner [6] examined the attached setup case. Glass and Potts [14] developed a two-phase method to find the optimal allocation of sublots. First, a powerful relaxation algorithm was derived to reduce the number of machines that needed to be considered. In the second stage, the critical path structure of an optimal solution is characterized. Steiner and Truscott [19] and Chen and Steiner [7] studied the no-wait problem, in which each subplot must be continuously processed from the start to its completion, without interruption between machines [15]. Biskup and Feldmann [2] presented an integer programming formulation to solve one-product flow shop problems. This model was then extended to deal with different settings and objectives, where variable sublots are considered. Feldmann and Biskup [10] extended the model to solve multi-product problems. In order to investigate the benefits of lot streaming, 160 small and medium sized instances were tested. Although the job shop scheduling problem is generally more complex compared to the flow shop problem, their model provided important inspiration for formulating the lot streaming problem in a job shop environment.

The job shop scheduling problem, on the contrary, has received little attention. Dauzère-Pérès and Lasserre [9] introduced an iterative procedure to solve the lot streaming problem in a job-shop environment. The procedure alternates between solving a lot-sizing problem with a given sequence of sublots on the machines and a standard job shop scheduling problem with fixed subplot sizes. By adopting the modified shifting bottleneck procedure [8], a good solution can be obtained within a few iterations. For the same problem, Buscher and Shen [3] presented an advanced tabu search algorithm which outperforms the previous heuristic. This algorithm is also able to reach the theoretical lower bounds for some hard benchmark instances in scheduling. In [17] a model for the lot streaming problem with setups was established. Aside from the makespan objective, cost-based measurements are integrated as well. Under the assumption

Udo Buscher is full professor of industrial management at the Dresden University of Technology in Germany. e-mail: udo.buscher@tu-dresden.de

Liji Shen is currently a research fellow at the Dresden University of Technology. e-mail: liji.shen@tu-dresden.de

that the subplot sizes are given, several examples were tested with the conclusion that equal-sized sublots provide better solutions in general.

The integer programming formulation presented in this paper is based on the studies of [4] and [17]. Necessary modifications are conducted in the first place. The model is then further developed to increase efficiency. Moreover, instead of employing fixed subplot sizes, test instances are solved with the determination of subplot sizes. According to our observation, optimal solutions are generally obtained with unequal-sized sublots, which obviously contradicts the assertion of [17].

The remainder of the paper is organized as follows: in the next section, an integer programming formulation for solving the lot streaming problem in a job shop production system with setups is developed. Various types of setups are incorporated in the model. Section 3 provides a detailed analysis of computational results. An illustrative example consisting of 3 jobs and 3 machines is first discussed. The computational results focusing on various aspects are then presented in detail. Brief conclusions are summarized in Section 4.

II. MODEL FORMULATION

A. Notations

Parameters:

A	set of pairs of operations constrained by precedence relations
n	total number of jobs
m	total number of machines
s	total number of sublots
i, i'	job indices, $i, i' = 1, \dots, n$
k, k'	machine indices, $k, k' = 1, \dots, m$
j, j'	subplot indices, $j, j' = 1, \dots, s$
D_i	demand of job i , i.e. the initial lot size
H	sufficiently large number
L	set of the last operations of sublots
M_k	machine k
O_{ijk}	the operation of the j th subplot of job i on machine k
p_{ik}^u	unit processing time of job i on machine k
r_{ik}	setup time of job i on machine k

Variables:

C_{max}	makespan
t_{ijk}	start time of operation O_{ijk}
X_{ij}	production quantity of the j th subplot of job i
δ_{ijk}	binary variable which equals 1 if setup is required before processing operation O_{ijk} ; 0 otherwise
$Y_{ijj'j'k}$	binary variable which equals 1 if Operation O_{ijk} is processed prior to Operation $O_{i'j'k}$; 0 otherwise.

B. Integer programming formulation

It is assumed that each job consists of m operations and must pass through each machine exactly once. All machines

are available at time zero. Furthermore, the total number of sublots is given and consistent subplot sizes are considered. In addition, transport times are negligible. With the notations and assumptions, the model can then be summarized as follows:

$$\min C_{max} \quad (1)$$

Subject to:

$$\sum_{j=1}^s X_{ij} = D_i \quad \forall i \quad (2)$$

$$X_{ij} \geq 0 \quad \forall i, j \quad (3)$$

$$\delta_{ijk} \leq X_{ij} \quad \forall i, j, k \quad (4)$$

$$t_{ijk'} \geq t_{ijk} + r_{ik} \cdot \delta_{ijk} + p_{ik}^u \cdot X_{ij} \quad \forall (O_{ijk}, O_{ijk'}) \in A \quad (5)$$

$$t_{i(j+1)k} \geq t_{ijk} + r_{ik} \cdot \delta_{ijk} + p_{ik}^u \cdot X_{ij} \quad \forall i, k, j < s \quad (6)$$

$$C_{max} \geq t_{isk} + r_{ik} \cdot \delta_{isk} + p_{ik}^u \cdot X_{is} \quad \forall i, O_{isk} \in L \quad (7)$$

$$t_{ijk} \geq t_{i'j'k} + r_{i'k} \cdot \delta_{i'j'k} + p_{i'k}^u \cdot X_{i'j'} - H \cdot Y_{ijj'j'k}$$

$$t_{i'j'k} \geq t_{ijk} + r_{ik} \cdot \delta_{ijk} + p_{ik}^u \cdot X_{ij} - H \cdot Y_{ijj'j'k} \quad (8)$$

$$Y_{ijj'j'k} + Y_{i'j'j'k} = 1 \quad \forall i \neq i', j, j' \quad (9)$$

$$\delta_{i1k} = 1 \quad \forall i, k \quad (10)$$

$$\delta_{i(j+1)k} \geq Y_{ijj'j'k} - Y_{i(j+1)i'j'k} \quad \forall i \neq i', j < s, j', k. \quad (10)$$

In our model we employ the conventional makespan objective function (1). Constraints (2) ensure that all required units are produced. Constraints (3) are the non-negativity conditions. Since subplot sizes may equal 0, the actual number of sublots is possibly smaller than the given number s . This adds flexibility to the formulation with the fixed total number of sublots (s). Obviously, no setup is necessary, if the corresponding subplot doesn't exist. Constraints (4) are therefore used to avoid redundant setups.

Constraints (5) represent the precedence relations of the operations that belong to the same subplot. In the model of [17] similar constraints are considered, which apply to the operations of different sublots as well. It should be pointed out that the operations of different sublots are not constrained by the precedence relation, since sublots are treated as separate jobs.

When attached setup times are taken into consideration, the setup of a certain machine cannot begin until the corresponding subplot has been transferred to this machine. Constraints (5) fulfil this requirement. On the other hand, detached setups can be performed in advance, with no regard to the availability of sublots. The constraints can then be slightly modified as:

$$t_{ijk'} + r_{i'k'} \cdot \delta_{ijj'k'} \geq t_{ijk} + r_{ik} \cdot \delta_{ijk} + p_{ik}^u \cdot X_{ij} \quad \forall (O_{ijk}, O_{ijj'k'}) \in A \quad (11)$$

As an illustration, both types of setups are depicted in figures 1 and 2, respectively.

Constraints (6) state that a subplot can only be scheduled on a certain machine after the sublots with smaller indices of the same job finish their processing. For instance, the second subplot can not be processed prior to the first subplot of the same job. Due to the simultaneous determination of subplot sequences and subplot sizes, constraints (6) can be employed without loss of generality. In the meantime, these constraints provide the basis for the concise formulation of setup times.

Constraints (7) indicate that the makespan is defined by the latest completion time of the last operation of the subplot with the maximal index (s). In the model developed in [17], the

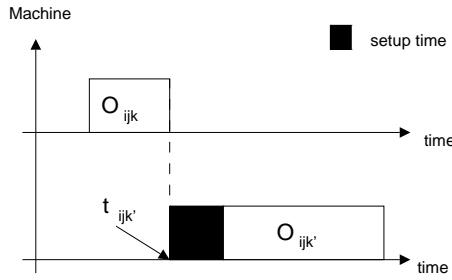


Fig. 1. Illustration of attached setups

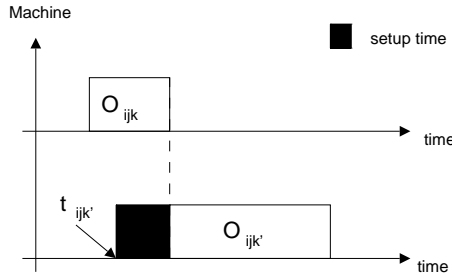


Fig. 2. Illustration of detached setups

makespan is similarly calculated, while the subplot precedence constraints (6) are neglected. This, however, is not always correct, since the s th sublots are not necessarily scheduled at the end.

Theoretically, constraints (6) can be removed. The makespan is thus determined by:

$$C_{max} \geq t_{ijk} + r_{ik} \cdot \delta_{ijk} + p_{ik}^u \cdot X_{ij} \quad \forall i, j, k. \quad (12)$$

Owing to the complex interaction between sublots and machines, this formulation generally requires more iterations to solve the identical problem.

Constraints (8) are adopted to determine the sequences on machines and to prevent overlapping of operations. If $Y_{ij'j'k}$ takes the value 1, only the first set of constraints is relevant, which indicates that operation $O_{i'j'k}$ must be processed after the completion of operation O_{ijk} . If $Y_{ij'j'k}$ equals 0, the second set of constraints operates in a similar manner.

In the model of [17], the last set of constraints in (8) is neglected. This, in the first place, contradicts the definition of the binary variable $Y_{ij'j'k}$. Moreover, setting both $Y_{ij'j'k}$ and $Y_{i'j'j'k}$ to be 1 leads to infeasible solutions.

In view of setups, constraints (9) ensure that the machines are set up before processing the first subplot of each job.

Note that only one setup is essential, if sublots of the same job are consecutively scheduled on a certain machine. In terms of Dauzère-Pérès and Lasserre [9], this is a scheduling problem with *sequence-dependent setup times*, which is difficult to solve. Instead of approximate modelling, constraints (10) formulate this situation precisely. According to (6), operation O_{ijk} should always be scheduled before $O_{i(j+1)k}$. If these two operations are processed directly one after the other, $\delta_{i(j+1)k}$ takes the value 0 automatically (see figure 3). As long as there is an operation of any other job in between, the right side of the corresponding inequation equals 1, which forces $\delta_{i(j+1)k}$ to be 1 (see figure 4). Therefore, constraints (10) ensure that all the consecutively scheduled sublots of the

same job are processed under a single setup. In the study

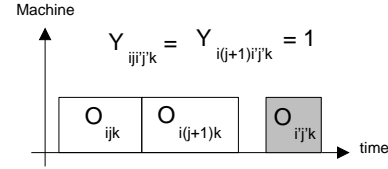
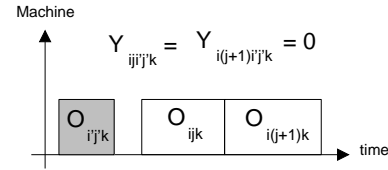


Fig. 3. Illustration of constraints (10) (I)

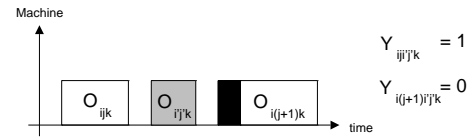


Fig. 4. Illustration of constraints (10) (II)

of [17] the setup requirements are written as:

$$s \cdot (Z_k - 1) \cdot \delta_{i(j+1)k} \geq \sum_{i'=1}^n \sum_{j'=1}^s (Y_{i(j+1)i'j'k} - Y_{ij'i'j'k}) \quad \forall j, k \text{ and } i \neq i',$$

where Z_k is the total number of operations to be processed on machine k . In comparison, we formulate this condition in a more intuitive and elegant way.

C. Extensions

1) *No-wait*: An important class of machine scheduling problems is characterized by a no-wait production system, where a job must be processed from start to completion without intermediate buffer between machines. In comparison to constraints (5), the no-wait requirement can be simply expressed as:

$$t_{ijk'} = t_{ijk} + r_{ik} \cdot \delta_{ijk} + p_{ik}^u \cdot X_{ij} \quad \forall (O_{ijk}, O_{ijk'}) \in A. \quad (13)$$

2) *Non-idling*: On the other hand, if a non-idling environment occurs, where all sublots of the same job must be continuously processed on a particular machine, constraints (6) can be modified as follows:

$$t_{i(j+1)k} = t_{ijk} + r_{ik} \cdot \delta_{ijk} + p_{ik}^u \cdot X_{ij} \quad \forall i, k \text{ and } j < s. \quad (14)$$

Obviously, only the setup before the first subplot of each job is required. The other binary variables related to setups are then equal to 0:

$$\delta_{ijk} = 0 \quad \forall i, k \text{ and } j \neq 1. \quad (15)$$

Moreover, only the sequence of jobs is relevant. Binary variables $Y_{ij'j'k}$ can thus be simplified as $Y_{i'j'k}$, which significantly reduces the complexity of the formulation. The

modified constraints concerning $Y_{ii'k}$ are summarized as follows:

$$\begin{cases} t_{i1k} \geq t_{i'sk} + r_{i'k} \cdot \delta_{i'sk} + p_{i'k}^u \cdot X_{i's} - H \cdot Y_{ii'k} \\ t_{i'1k} \geq t_{isk} + r_{ik} \cdot \delta_{isk} + p_{ik}^u \cdot X_{is} - H \cdot Y_{i'ik} \\ Y_{ii'k} + Y_{i'ik} = 1 \quad \forall i, k \text{ and } i' \neq i. \end{cases} \quad (16)$$

In order to prevent overlapping on machines, we can take advantage of the attribute of the non-idling case and only need to compare the start time of the first subplot of a certain job with the completion time of the last subplot of another job.

3) *Non-intermingle*: Another situation especially associated with the lot streaming problem is the non-intermingling setting. This case requires that no interruption from any other job is allowed while processing a particular job. Therefore, constraints (14) are the sufficient but not necessary conditions, whereas (15) and (16) must be fulfilled.

4) *Special constraints*: According to the subplot precedence constraints (6), operation $O_{i'j'k}$ should be scheduled prior to operation $O_{i'(j'+1)k}$. Figure 5 illustrates all possible positions of a third operation O_{ijk} . The corresponding values of the binary variables $Y_{ijj'j'k}$ are listed in table I.

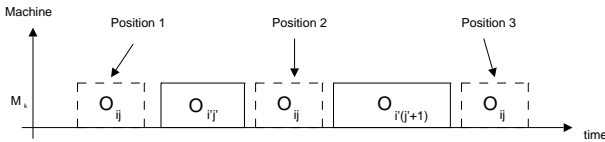


Fig. 5. Illustration of condition (17)

TABLE I
ILLUSTRATION OF CONDITION (17)

	Position 1	Position 2	Position 3
$Y_{ijj'j'k}$	1	0	0
$Y_{ijj'(j'+1)k}$	1	1	0

The following constraints describe these attributes of $Y_{ijj'j'k}$:

$$Y_{ijj'j'k} \leq Y_{ijj'(j'+1)k} \quad \forall i, j, j' < s \text{ and } i' \neq i. \quad (17)$$

According to figure 6 and table II, the other constraints

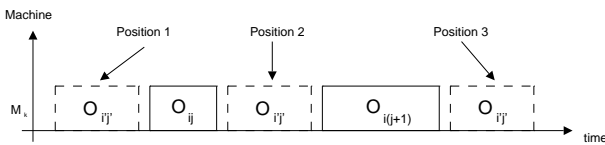


Fig. 6. Illustration of condition (18)

TABLE II
ILLUSTRATION OF CONDITION (18)

	Position 1	Position 2	Position 3
$Y_{ijj'j'k}$	0	1	1
$Y_{i(j+1)i'j'k}$	0	0	1

concerning $Y_{ijj'j'k}$ can be expressed as:

$$Y_{ijj'j'k} \geq Y_{i(j+1)i'j'k} \quad \forall i, j', j < s \text{ and } i' \neq i \quad (18)$$

Paper [10] contains similar constraints to solve lot streaming flow shop problems. The function of these constraints in a complex job shop environment will be discussed in the next section.

III. COMPUTATIONAL RESULTS

The integer programming formulation addressed in the previous section was implemented in the optimization software Lingo 9.0 on a personal computer (Athlon 64 X2 4800+, 2450MHZ). In this section an illustrative example composed of 3 jobs and 3 machines is first discussed. The computational results focusing on various aspects are then presented in detail.

A. An illustrative example

First, we consider an example with 3 jobs and 3 machines. All jobs consist of 3 operations and each of them is to be processed on a unique machine. Unit processing times and demands, as well as flow patterns of machines are shown in tables III and IV. The setup times of operations are set to be 50% of their unit processing times. Moreover, the total number of sublots is given as 3.

TABLE III
DEMANDS AND UNIT PROCESSING TIMES OF THE ILLUSTRATIVE EXAMPLE

	Demand D_i	Unit processing time		
		$k = 1$	$k = 2$	$k = 3$
$i = 1$	12	30	30	20
$i = 2$	24	30	20	30
$i = 3$	36	30	40	10

TABLE IV
FLOW PATTERNS OF MACHINES

i	Flow pattern		
$i = 1$	M_1	M_2	M_3
$i = 2$	M_2	M_3	M_1
$i = 3$	M_2	M_1	M_3

This 3-3 problem was first solved without the application of the lot streaming strategy. The associated schedule with the makespan of 3420 is shown in figure 7.

The problem is then solved taking account of lot streaming. With regard to attached setups, the optimal subplot sizes obtained are :

$$\begin{aligned} X_{11} &= 3 & X_{12} &= 5 & X_{13} &= 4 \\ X_{21} &= 10 & X_{22} &= 7 & X_{23} &= 7 \\ X_{31} &= 17 & X_{32} &= 13 & X_{33} &= 6. \end{aligned}$$

The corresponding schedule with the makespan of 2435 is illustrated in figure 8. In the case of detached setup times, the optimal subplot sizes are as follows:

$$\begin{aligned} X_{11} &= 4 & X_{12} &= 5 & X_{13} &= 3 \\ X_{21} &= 9 & X_{22} &= 7 & X_{23} &= 8 \\ X_{31} &= 17 & X_{32} &= 13 & X_{33} &= 6. \end{aligned}$$

The associated optimal schedule is depicted in figure 9.

Since the setup for operation O_{311} starts earlier, it can be processed immediately after the completion of its previous operation O_{312} . The same applies to operation O_{213} and O_{313} as well. Consequently, the makespan is further improved

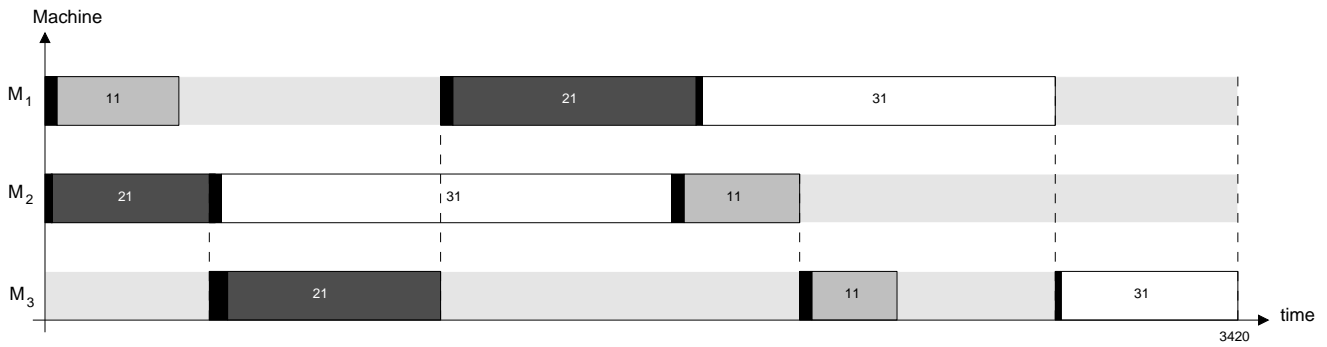


Fig. 7. Solution to the example (without lot streaming)

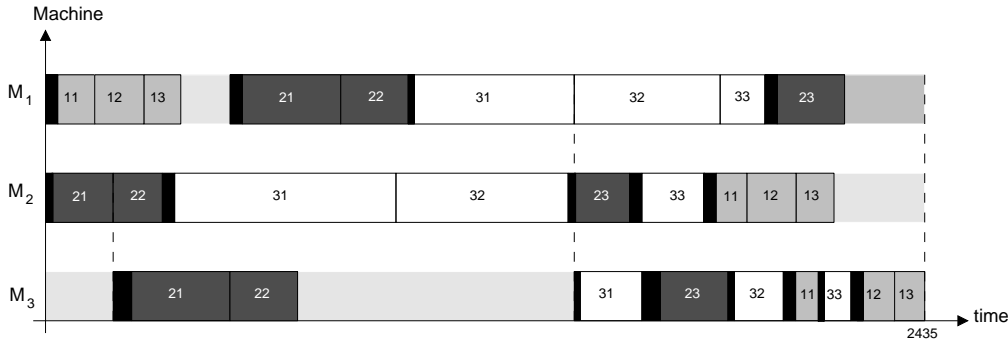


Fig. 8. Solution to the example (with lot streaming and attached setups)

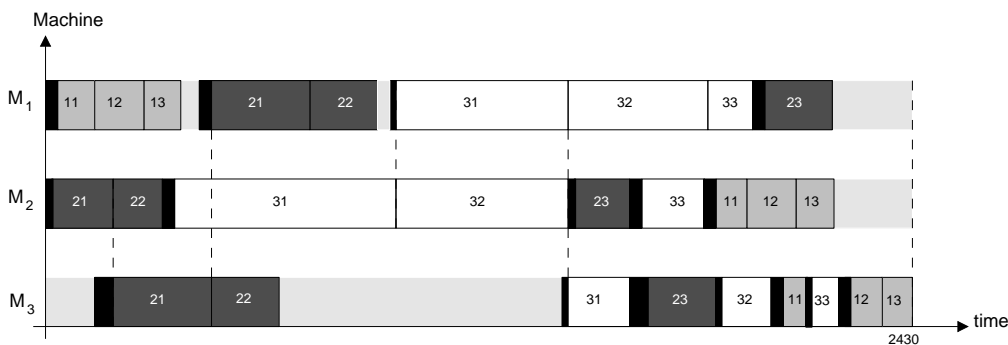


Fig. 9. Solution to the example (with lot streaming and detached setups)

by 5 units due to an additional reduction of idling time of machine 3. In comparison to the schedule obtained without the consideration of lot streaming, the improvement of makespan is 28.80% concerning attached setups and 28.95% in the case of detached setups. With the application of the lot streaming strategy, the machine utilization becomes much more consistent. Therefore, the makespan is improved to a considerable extend.

Figures 10 and 11 show the optimal schedules subject to no-wait and no-idling assumption, respectively. We can clearly see that, due to the specific requirements, the resulting schedules are adjusted accordingly, which lead to longer makespan.

In order to investigate the influence of setup times on the performance of lot streaming techniques, this 3-3 problem with different setup times is solved. It is first assumed that all setup times are negligible. All setup times are then set to be very large (1000). As shown in figure 12, in the absence of setup times, intermingling among the sublots of different jobs becomes more frequent. As setup times increase, the

sublots of the same job tend to be scheduled successively, so that no additional setup is required. Along with the effect of operation overlapping at different stages, the makespan is remarkably improved. This result suggests that lot streaming is still advantageous in spite of the existence of extreme setup times.

B. Benefit of lot streaming

By the implementation of Lingo 9.0, however, only small instances of job shop problems can be solved optimally. While solving a 2-2 problem with 4 sublots requires less than 2 seconds, the optimal solution for a 3-3 problem with 4 sublots cannot be obtained within 6 hours.

In our study 96 instances of job shop problems, which consist of 2 to 3 jobs and 2 to 6 machines, are tested. Each of the instances with 2 jobs is solved by adopting 1 to 4 sublots. The instances containing 3 jobs are solved employing 1 to 3 sublots. The mean improvement of makespan is summarised in table V.

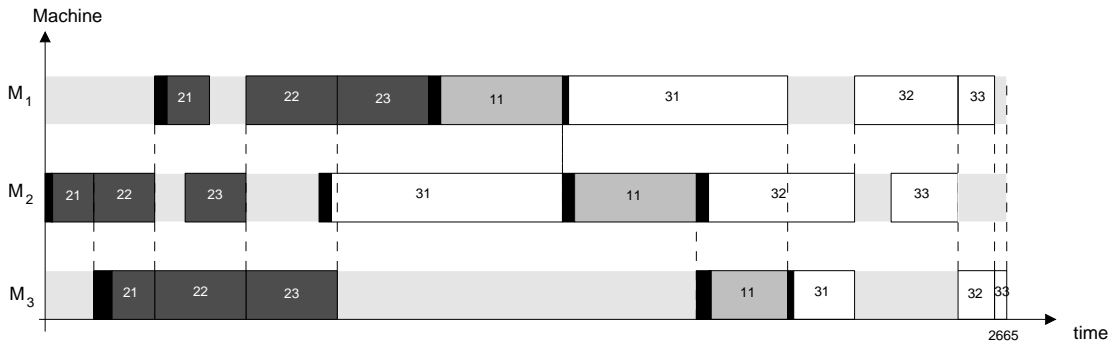


Fig. 10. Solution to the example (subject to no-wait requirement)

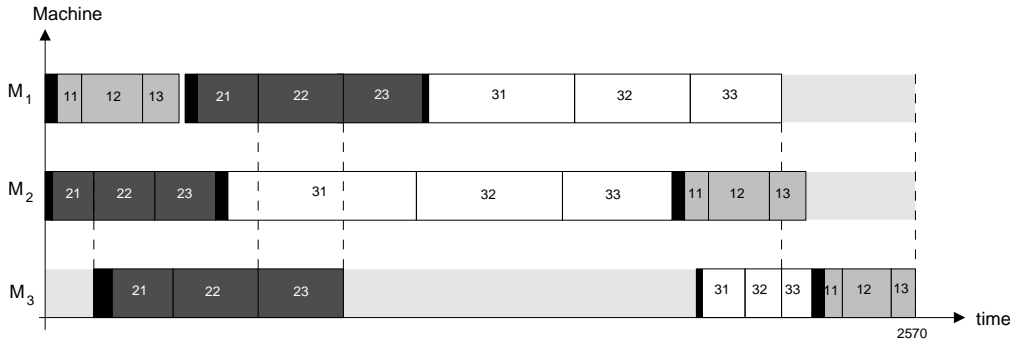


Fig. 11. Solution to the example (subject to no-idling requirement)

TABLE V
MEAN IMPROVEMENT OF MAKESPAN

job-machine	2 sublots		3 sublots		4 sublots	
	Improvement*	Percentage*	Improvement	Percentage	Improvement	Percentage
2-2	6.04%	100.00%	0.00%	0.00%	0.00%	0.00%
2-3	20.61%	71.03%	5.98%	20.84%	2.39%	8.13%
2-4	25.02%	63.54%	9.82%	25.00%	4.53%	11.46%
2-5	32.89%	69.45%	10.08%	21.28%	4.40%	9.26%
2-6	32.04%	67.06%	11.12%	23.26%	4.65%	9.69%
3-3	23.08%	77.88%	6.92%	22.12%	-	-
Mean	23.44%	74.83%	7.32%	18.75%	3.19%	7.71%

* Improvement and percentage are calculated by: $(C_{max,s-1} - C_{max,s}) / C_{max,1}$ and $(C_{max,s-1} - C_{max,s}) / (C_{max,1} - C_{max,4})$, respectively. $C_{max,s}$ represents the makespan obtained by applying s sublots.

The average improvement of makespan amounts to 23.44% by applying 2 sublots. With regard to 3 sublots, the benefit of lot streaming is abruptly reduced to 7.32%. In general, 74.83% of potential makespan reduction is already achieved by the application of 2 sublots, whereas the proportion of 4 sublots is marginal. This observation suggests employing merely 2 sublots, so that the most advantage of lot streaming can be obtained while saving computing time.

C. Impact of setup times

In order to analyze the relationship between setup times and makespan reduction, we adopt the settings from [9]. The setup times are set to be 1%, 10%, 50%, 100%, 200% and 400% of the unit processing times. The computational results of 96 test instances are listed in table VI.

As plotted in figure 13, while the proportion between setup times and processing times rises, the advantage of lot streaming declines in accordance. Nevertheless, the improvement of makespan with 2 sublots exceeds 20%.

Evidently, there is a trade off between the time saved by splitting into sublots and the extra time required due to

additional setups. Although the size of setup times imposes a negative influence on the reduction of makespan, lot streaming is still unnegligible efficient.

D. Solutions with equal-sized sublots

As mentioned in the first section, many researches involved examining the performance of equal-sized sublots. In this respect, constraints (2) are modified as:

$$X_{ij} = D_i/s \quad \forall i, j \quad (19)$$

In comparison to solving the problem optimally, the subplot sizes are predetermined, which significantly reduces the complexity of the problem. As a result, Lingo program can be remarkably accelerated. In our study, the test instances are also solved by adopting equal-sized sublots. As presented in table VII, the deviation of makespan is surprisingly only 3.05% on average, while the necessary iterations to solve identical problems fall sharply. This valuable information suggests that we can take advantage of the trade off between the reduction of computing time and the increment of makespan to solve larger instances of job shop problems.

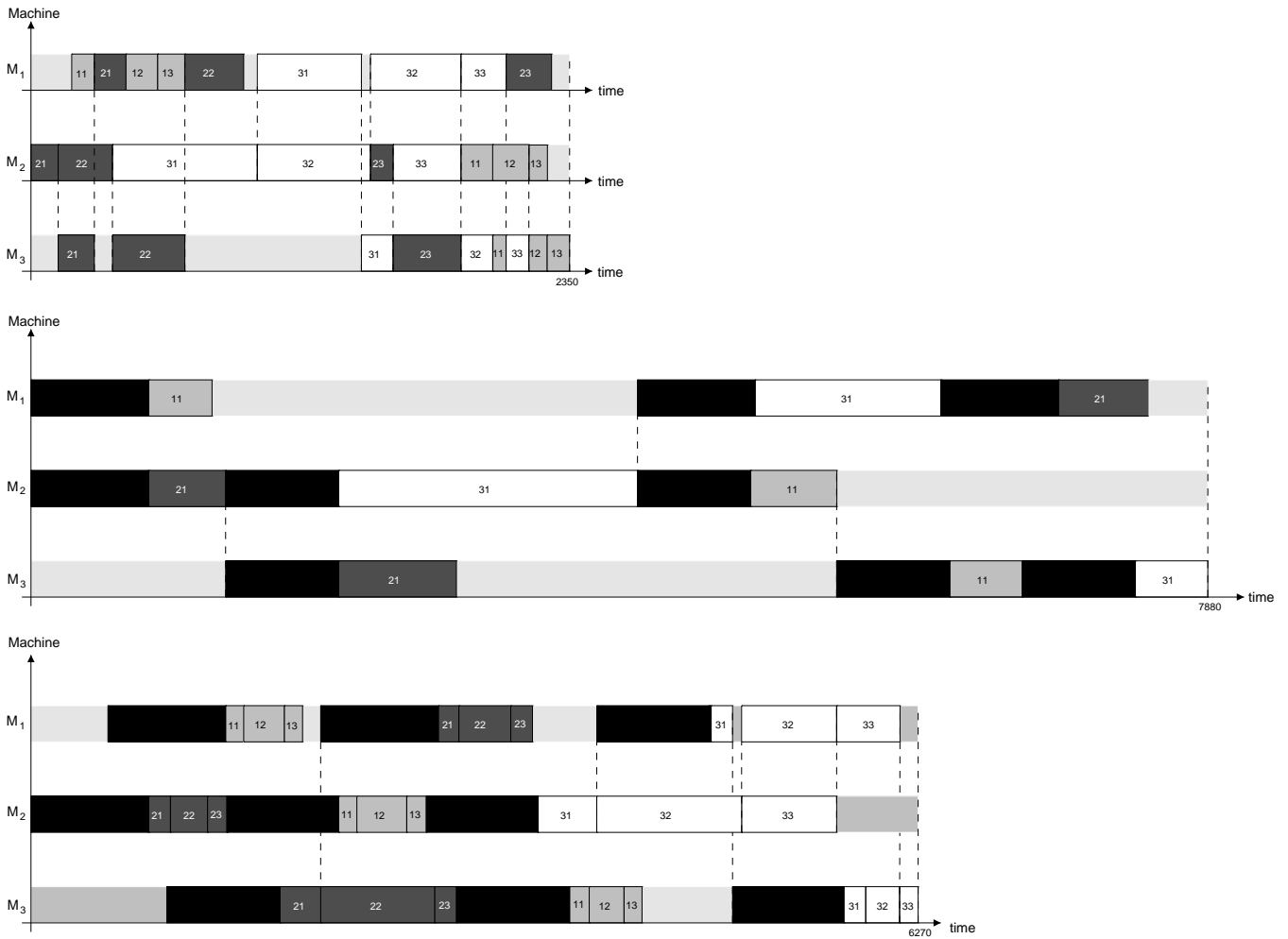


Fig. 12. Solutions of the example (with different setup times)

TABLE VI
PERFORMANCE OF LOT STREAMING WITH VARIOUS SETUP TIMES

Setup time	2 sublots		3 sublots		4 sublots	
	Improvement*	Percentage*	Improvement	Percentage	Improvement	Percentage
1%	24.49%	74.38%	7.61%	18.69%	3.56%	8.33%
10%	24.28%	74.17%	7.61%	18.74%	3.61%	8.51%
50%	23.61%	74.61%	7.38%	18.65%	3.38%	8.09%
100%	23.28%	74.79%	7.37%	18.99%	3.06%	7.47%
200%	22.75%	74.90%	7.03%	18.49%	3.20%	7.93%
400%	22.21%	76.10%	6.92%	18.95%	2.36%	5.93%

* Improvement and percentage are calculated by: $(C_{max,s-1} - C_{max,s})/C_{max,1}$ and $(C_{max,s-1} - C_{max,s})/(C_{max,1} - C_{max,4})$, respectively. $C_{max,s}$ represents the makespan obtained by applying s sublots.

In consequence, satisfying solutions can be obtained within realistic time.

E. Evaluation of the formulation

One main difference of our formulation compared to the model proposed in [17] is the successful removal of the operation index. In order to compare the efficiency of these two formulations, we implemented their model in Lingo 9.0 as well (after necessary corrections, so that feasible solutions are generated). 45 instances were tested under identical circumstances.

As shown in table VIII, our formulation requires significantly less iterations for most of the instances. This

advantage becomes especially obvious when the problem size increases. The experiment confirms that our formulation is not only straightforward but also more efficient in general. In our model constraints (17) and (18) are incorporated to describe attributes of the binary variable $Y_{ijj'jk}$. Constraints similar to (17) are also considered in [10]. In terms of Feldmann and Biskup [10], the number of iterations required to solve a flow shop problem could be reduced to 60% compared to the model without these restrictions. However, the scheduling reality in a job shop environment is much more complicated. In order to investigate the function of these constraints, 60 instances were tested. The results are depicted in figure 14, where the coefficient is calculated as

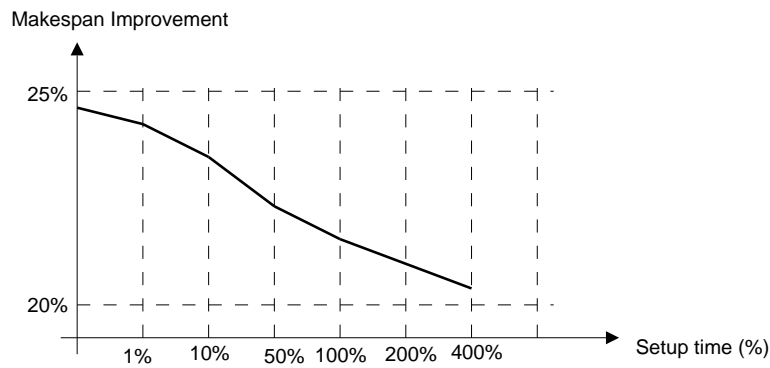


Fig. 13. Makespan improvement with different setup times

TABLE VII
PERFORMANCE OF EQUAL-SIZED SUBLOTS

job· machine	2 sublots	3 sublots	4 sublots	Percentage *
2-3	4.11%	4.75%	4.40%	16,68%
2-4	3.40%	4.37%	4.44%	18,11%
2-5	5.78%	7.41%	7.48%	3,65%
2-6	3.09%	4.11%	4.76%	2,50%
3-3	1.91%	2.41%	-	4,94%
Mean	3.05%	3.84%	4.21%	8,19%

$$* \text{ Percentage} = \frac{\text{iteration required applying equal sublots}}{\text{iteration required applying consistent sublots}}$$

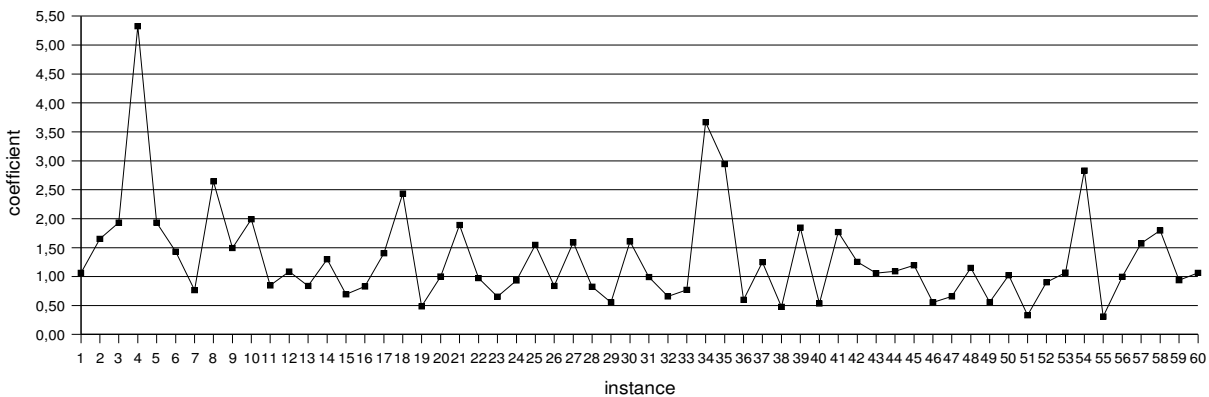


Fig. 14. Performance of (17) and (18)

follows:

$$\text{coefficient} = \frac{\text{Iterations required without (17) and (18)}}{\text{Iterations required employing (17) and (18)}}$$

By employing these constraints, the necessary iterations for some instances, on the one hand, can be reduced to less than 50%. On the other hand, solving some instances with these constraints demands exceedingly more iterations. Unlike in a flow shop environment, no conclusive behaviour pattern of these constraints was recognizable.

IV. CONCLUSION

This paper addresses the lot streaming problem in a job shop environment, where setup times are included. The proposed integer programming formulation sufficiently describes the processing dynamics of individual sublots and enables the simultaneous determination of schedules on machines and subplot sizes. In view of setup times, various types of setups are incorporated. The model is then further

developed to fulfil the requirements of special production systems.

Computational results confirm that the makespan can be considerably improved, when lot streaming techniques are applied to the standard job shop problem. Furthermore, detailed analysis is conducted to reveal the relation between setup times and makespan reduction. Although the improvement of makespan declines as setup times increase, lot streaming is still advantageous.

In comparison to the model established in [17], our formulation is not only straightforward but also more efficient in general.

However, by the implementation of the optimization-based software Lingo 9.0, only small instances of the job shop problem can be optimally solved within a realistic time span. Thus, the development of effective heuristics to solve large instances of the problem is desirable for future study. For instance, the implementation of metaheuristic is advisable.

TABLE VIII
COMPARISON OF THE PERFORMANCE OF TWO FORMULATIONS

Problem size	(a) Iter. (our model)	(b) Iter. (model of Low et al)	Percentage (a/b)	
3-3-2	1	339311	1661149	20,43%
	2	10587	10821	97,84%
	3	14750	28919	51,00%
	4	27910	56208	49,65%
	5	18304	31785	57,59%
	6	15267	49228	31,01%
	7	18750	42579	44,04%
	8	11069	11349	97,53%
	9	11169	11494	97,17%
	10	5494	4707	116,72%
	11	17313	24313	71,21%
	12	12117	15198	79,73%
	13	14312	20909	68,45%
	14	18616	20183	92,24%
	15	37712	24313	155,11%
	16	21315	27663	77,05%
	17	21678	14775	146,72%
	18	21778	43153	50,47%
	19	19988	33179	60,24%
			77,06%	
2-2-4	20	16485	45645	36,12%
	21	9829	13330	73,74%
	22	36019	52291	68,88%
	23	18457	34088	54,15%
	24	31726	72561	43,72%
	25	33657	37683	89,32%
	26	17932	18015	99,54%
	27	19652	24281	80,94%
	28	16574	30150	54,97%
	29	5412	9431	57,39%
	30	4680	10576	44,25%
	31	14248	22067	64,57%
	32	16422	17778	92,37%
	33	2512	5477	45,86%
	34	3361	17367	19,35%
			61,68%	
3-3-3	35	763827	950003	80,40%
	36	2301475	5993093	38,40%
	37	5255761	12591491	41,74%
	38	3162333	8806108	35,91%
	39	2229550	9719360	22,94%
	40	769311	4785587	16,08%
	41	1033909	1615861	63,99%
	42	739141	2404653	30,74%
	43	656268	1410262	46,54%
	44	173255	2248802	7,70%
	45	80564	131995	61,04%
			40,50%	

REFERENCES

- [1] K.R. Baker. Lot Streaming in the Two-Machine Fflow Shop with Setup Times, *Annals of Operations Research*, vol. 57, pp.1-11, 1995.
- [2] D. Biskup and M. Feldmann. Lot Streaming with Variable Sublots: An Integer Programming Formulation, *Journal of Operational Research Society*, vol. 57, pp. 296-303, 2006.
- [3] U. Buscher and L. Shen. An Integrated Tabu Search Algorithm for the Lot Streaming Problem in Job Shops, *European Journal of Operational Research*, vol. 19, pp. 385-399, 2009.
- [4] U. Buscher and L. Shen. An Integer Programming Formulation for the Lot Streaming Problem in a Job Shop Environment with Setups, *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2011*, IMECS 2011, 16-18 March, 2011, Hong Kong, pp. 1343-1348.
- [5] J.Chen and G. Steiner. Lot streaming with detached setups in three-machine flow shops, *European Journal of Operational Research* vol. 96, pp. 591-611, 1996.
- [6] J. Chen and G. Steiner. Lot Streaming with Attached Setups in Three-Machine Flow Shops, *IIE Transactions* 30, pp. 1075-1084, 1998.
- [7] J. Chen and G. Steiner. On Discrete Lot Streaming in No-Wait Flow Shops, *IIE Transactions*, vol. 35, pp. 91-101, 2003.
- [8] S. Dautère-Pères and J.B. Lasserre. A modified shifting bottleneck procedure for job-shop scheduling, *International Journal of Production Research*, vol. 31, pp. 923-932, 1993.
- [9] S. Dautère-Pères and J.B. Lasserre. Lot Streaming in Job-Shop Scheduling, *Operations Research*, vol. 45, pp. 584-595, 1997.
- [10] M. Feldmann and D. Biskup. On Lot Streaming with Multiple Products. Discussion paper, No. 542, Department of Business Administration and Economics, Bielefeld University, Germany 2005.
- [11] S. French. In sequencing and scheduling: an introduction to the mathematics of the job-shop. Horwood: Chichester, U.K. 1982.
- [12] M.R. Garey, D.S. Johnson and Ravi Sethi. The complexity of flowshop and jobshop scheduling, *Mathematical methods of Operations Research*, vol. 1, pp. 117-129, 1976.
- [13] C.A. Glass, J.N.D. Gupta and C.N. Potts. Lot streaming in three-stage production processes. *European Journal of Operational Research*, vol. 75, pp. 378-394, 1994.
- [14] C.A. Glass and C.N. Potts. Structural properties of lot streaming in a flow shop. *Mathematical methods of Operations Research* vol. 23, pp. 624-639, 1998.
- [15] N.G. Hall and C. Sriskandarajah. A survey of machine scheduling problems with blocking and no-wait in process, *Operations Research* vol. 44, pp. 510-525, 1996.
- [16] S.M. Johnson. Optimal two- and three-stage production schedules with set-up times included. *Naval Research Logistics Quarterly*, vol. 1, pp. 61-68, 1954.
- [17] C. Low, C.M. Hsu and K.I. Huang. Benefits of Lot Splitting in Job-Shop Scheduling, *International Journal of Advanced Manufacturing Technology*, vol. 24, pp. 773-780, 2004.
- [18] C.N. Potts and K.R. Baker. Flow Shop Scheduling with Lot Streaming, *Operations Research Letters*, vol. 8, pp. 297-303, 1989.
- [19] G. Steiner and G.W. Truscott. Batch scheduling to minimize cycle

- time, flow time and processing cost. *IIE Transactions*, vol. 25, pp. 90-97, 1993.
- [20] D. Trietsch and K.R. Baker. Basic techniques for lot streaming. *Operations Research*, vol. 41, pp. 1065-1076, 1993.
- [21] R.G. Vickson and B. E. Alfredsson. Two- and Three-Machine Flow Shop Scheduling Problems with Equal Sized Transfer Batches, *International Journal of Production Research*, vol. 30, pp. 1551-1574, 1992.