Proceedings of the 13th
Mediterranean Conference on Control and Automation
Limassol, Cyprus, June 27-29, 2005

Lot Streaming Technique in Job-shop Environment

Felix T. S. Chan*, T. C. Wong, and Philip L. Y. Chan, Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong

Abstract — **Applying Lot Streaming (LS) technique, a new approach is applied to determine LS conditions in Job-shop Scheduling Problem (JSP) using Genetic Algorithms (GA). LS approach allows a job (lot) to be split into a number of smaller sub-jobs (sub-lots) so that successive operations of the same job can be overlapped. In this connection, the total completion time of the whole job may be shortened. The proposed approach is developed to solve two sub-problems. The first sub-problem is called LS problem in which LS conditions are determined and another sub-problem is JSP after LS conditions are determined. A number of benchmarked problems will be investigated to study the optimum LS conditions in Just-In-Time (JIT) environment. Experiment results suggest that the model works fairly well with different objectives and good solutions can be obtained within reasonable time frame.**

I. Introduction

For traditional Job-shop Scheduling Problem (JSP), jobs are processed on machines in different orders. The problem is to sequence jobs on each machine to optimize objectives like makespan, earliness, lateness, etc. Given (1…m) machines and (1…n) jobs, the total number of possible solutions is $(n!)^m$. Thus, this problem is NP-hard. Over years, issues related to JSP are prevalent. One common assumption of classical JSP is that jobs (lots) cannot be split into smaller jobs (sub-jobs or sub-lots) due to single lot size [1-2]. If it is relaxed, non-single unit jobs can be split. However, this will increase the total setup time between distinct jobs and sub-jobs. Thus, extra setup may lengthen the overall schedule time. On the other hand, job splitting can improve the schedule by decreasing the number of late jobs and the total lateness. In this paper, the objective is the function of overall penalty cost and total setup cost (time). Overall penalty cost (OPC) is defined as the sum of earliness cost per hour early per unit of early jobs and lateness cost per hour late per unit of late jobs. Total setup cost (TSC) is the product of machining cost per hour of machines and total setup time during fixture changeover between lots (or sub-lots) on machines. “Fixture” is used to secure the jobs on the machine for completing the operations and each job must have a unique fixture. The cost of splitting a lot can also be justified by the TSC. In this research, the objective is to minimize the objective value by determining the optimum LS conditions which are defined as the determination of (i) the split lots (which lots to be split), (ii) the split number (the number of sub-lots of each lot), and (iii) the split size (the size of each sub-lot of each lot). The current study thus is trying to address the optimum LS conditions to JSP using Genetic Algorithms (GA).

*Felix. T. S. Chan is with the University of Hong Kong, Pokfulam Rd., Hong Kong (phone: (852) 28597059; fax: (852) 28586535; e-mail: ftschan@hkucc.hku.hk).

II. Literature Review

In reality, it is a common practice to apply LS to improve the productivity of a manufacturing workshop. For instance, most of the LS approaches are dedicated to Flow-shop Scheduling Problem (FSP) [3-5]. For traditional FSP, lots are processed in the same order. Therefore, it is fruitful to split lots for expediting the production process. [3] present a new heuristic method for minimizing the idle time on bottleneck machine to equal size LS in FSP. [4] address LS in 3 elements, i.e. the number of sub-lots, sub-lot size and sub-lots processing sequence. [5] propose a Hybrid Genetic Algorithm which incorporates Linear Programming and a Pair-wise Interchange method for LS in FSP. [6] study LS in Open-shop Scheduling Problem (OSP) which processing sequence of lots is unimportant.

In fact, the benefits of LS to FSP are much more obvious than that of LS to JSP. However, the application of LS to JSP is somewhat insufficient. From Figure 1, it shows that the application of LS can improve the overall schedule time in job-shop. It is assumed that the processing sequence of sub-lots follows the sequence of its original lot. In general, there are 4 types of LS approaches to general scheduling problems including (I) Equal size sub-lots without intermittent idling does not allow idle time between sub-lots on the same machine, (II) Equal size sub-lots with intermittent idling, (III) Varied size sub-lots without intermittent idling, and (IV) Varied size sub-lots with intermittent idling. For detailed description, please refer to the work done by [7]. The current manufacturing problem allows idle time between lots and sub-lots on the same machine, thus types I and III will not be considered. To examine the performance differences between equal and varied LS types with idle time, LSGAei and LSGAvi will be developed respectively. Then comparisons will be carried out on these two models. Moreover, it is assumed that once a lot is split, the number and size of its sub-lots are fixed throughout the schedule.

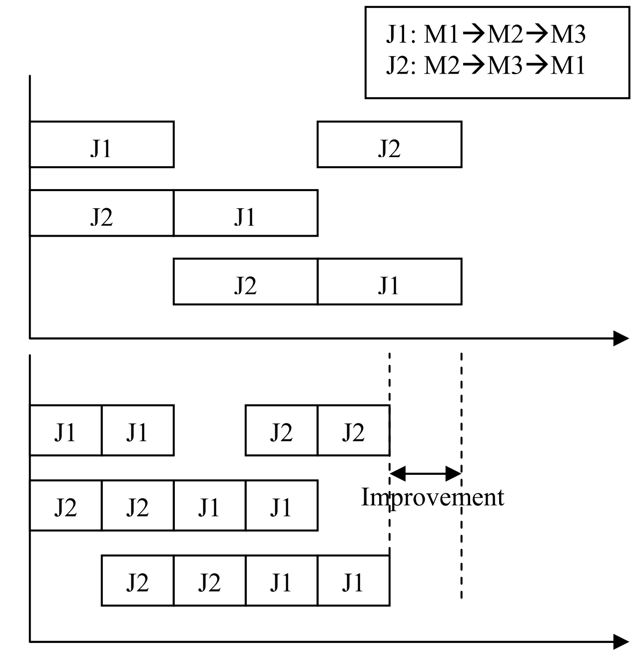

Fig.1. A job-shop problem without (upper) and with (lower) LS

Despite of limited LS applications to JSP, some have attempted to address the benefits. [8] present an iterative procedure to determine the sub-lot size for a given sub-lot sequence and JSP with fixed sub-lot size using LP and GA. They have shown that results near to lower bound can be obtained, but no attention is given to the selection of split lots. [2] study a lot splitting heuristic for JSP in dynamic environment. Particularly, the importance of the pre-setup time to the schedule performance is highlighted. However, they did not explicitly explain the impact of different dynamic factors on this pre-setup time. If jobs are heavily delayed, pre-setup may lengthen the overall schedule time. Therefore, pre-setup is not considered here.

This paper will be organized as follows. The mechanism of the proposed methodology will be explained in the next section. In section 4, computational results will be investigated to examine the performance of the proposed method. An illustrative example will be studied in section 5. Finally, conclusions will be drawn together with future research direction.

III. THE PROPOSED ALGORITHM

A. Model Notations

TABLE 1
MODEL NOTATIONS

Symbols	Descriptions
LSGAei	Model with equal size LS with idle time
LSGAvi	Model with varied size LS with idle time
W1	Weightings on overall penalty cost
W2	Weightings on total setup cost
m	Total number of machines
n	Total number of original lots
n'	Total number of sub-lots
J_i	Job i
J_{ij}	j^{th} lot of J_i
S_i	Number of sub-lots of J_i
F_i	Fixture of J_i
L_i	Original lot size of J_i
Q_{ij}	Quantity of J_{ij}
MS_{ik}	k^{th} machine for J_i
Pt_{ik}	Processing time on k^{th} machine of J_i
St_{ijk}	Start time of J_{ij} on machine k
C_{ij}	Completion date of J_{ij}
D_i	Due date of J_i
su_k	Total setup time on machine k
ec_i	Earliness cost per hour per unit of early J_i
tc_i	Tardiness cost per hour per unit of late J_i
mc_k	Machining cost of machine k per hour

B. Model Formulations

The proposed model is trying to solve two sub-problems, Sub-Problem one (SP1): Determination of LS conditions and Sub-Problem two (SP2): Solving JSP after LS conditions are determined. Equation (1) is used to evaluate each pair of SP1 and SP2 solutions. Constraint (2) requires the sum of all sub-lot sizes should satisfy the original lot size. (3) ensures that the processing sequence of sub-lots corresponds to the predetermined order. Equations (4)-(6) specify the range of variables i, j, and k. Also, each machine can process at most one job and all variables ≥ 0.

Min. Objective Value (OV)
= W1 x OPC + W2 x TSC

$$W1\times\left(\sum_i\sum_j(a_{ij}\times ec_i + b_{ij}\times tc_i)\times Q_{ij}\right) + W2\times\left(\sum_k su_k\times mc_k\right)$$

(1)

where
If $C_{ij} < D_i$, $a_{ij} = D_i - C_{ij}$ and $a_{ij} = 0$.
If $C_{ij} > D_i$, $b_{ij} = C_{ij} - D_i$ and $a_{ij} = 0$.
If $C_{ij} = D_i$, $a_{ij} = b_{ij} = 0$.

$$\sum_j Q_{ij} - L_i = 0 \quad \forall\, i \tag{2}$$

$$St_{ijMS_{i(k+1)}} \geq Pt_{ik} \times Q_{ij} + St_{ijMS_{ik}} \quad \forall\, i, j \tag{3}$$

$$1 \leq i \leq n \tag{4}$$

$$1 \leq j \leq S_i \tag{5}$$

$$1 \leq k \leq m \tag{6}$$

C. Genetic Algorithms

GA is a kind of evolutionary optimization methods as proposed by [9]. Its principle is based on the natural evolution. In terms of GA, solutions are encoded in so-called chromosome to form a solution pool. Each chromosome in the pool is evaluated based on the objective function to obtain the FITNESS VALUE. Applying the rule of survival-of-the-fittest, chromosomes with higher fitness values will have more chances to survive. It means that good chromosomes will either proceed to the new pool or be combined to generate new chromosomes. This process is called CROSSOVER. For extensive review on genetic crossover operators, please refer to [10]. These new chromosomes then may be self-tuned called MUTATION with a probability MUTATION RATE (MR). Each pool then represents a GENERATION, thus the procedure will continue to run until the terminating criteria are met. The best solution is then obtained at the final stage. Noted that the size of solution pool refers to POPULATION SIZE (PS) and the total number of pools defines the MAXIMUM NUMBER OF GENERATIONS (GEN).

D. Lot Streaming Technique

1) Development of LSGAei

SP1 is to determine the LS conditions. For equal size LS with idle time, LSGAei is developed. Using GA, a solution to SP1 is defined as a chromosome of size n and each gene represents S_i. With respect to equal size LS, $Q_{ij} = L_i/S_i$ subject to $S_i \leq L_i$. In some cases, if L_i/S_i is not an integer, the last sub-lot of J_i equals to $L_i - \sum Q_{ij}$ for j = 1…S_i-1. After splitting n lots to n' sub-lots, the next step is to solve JSP with n' independent lots, i.e. SP2. The scheduling results will be used to evaluate the SP1 chromosome. The objective value will be transformed into the fitness value using equation (7). Then LSGAei runs until terminating criteria are met. Good chromosomes will be chosen to perform crossover operation. For LSGAei, a simple 2-cut-point crossover (2X) operator is implemented. To implement 2X, two random cut points "|" are generated. The genes enclosed by the cut points are interchanged between two chromosomes to form two new offsprings. Then these new offsprings will perform mutation operation subject to a MR. Mutation operation is defined as the re-assignment of S_i value to the genes. The evolutionary scheme is that only offsprings will enter the new generation. If the offsprings are illegal, the crossover and mutation operations will be re-executed to the parental chromosomes until feasible offsprings are obtained.

$$Fitness\ Value(FV) = \frac{MAX - OV + MIN}{AVERAGE} \tag{7}$$

MAX→the maximum OV of the same generation
MIN→the minimum OV of the same generation
AVERAGE→the average OV of the same generation

2) Development of LSGAvi

For varied size LS with idle time, LSGAvi is developed. Applying LSGAvi, the string definition is (X, Y) as shown in Figure 2. (1, i) gives the same solution as LSGAei representing S_i for all lot i. (j, i) defines the size of each sub-lot of lot i where j = 2 … S_i + 1. 2X operation and mutation are illustrated in Figures 2a and 2b respectively. The main differences between LSGAei and LSGAvi are that the latter can determine the split lots, the split number, and the split size at the same time, but the former can only determine the split lots and the split number with equal split size. The fitness value of each chromosome is obtained using equation (7). The evolutionary scheme is the same as LSGAei.

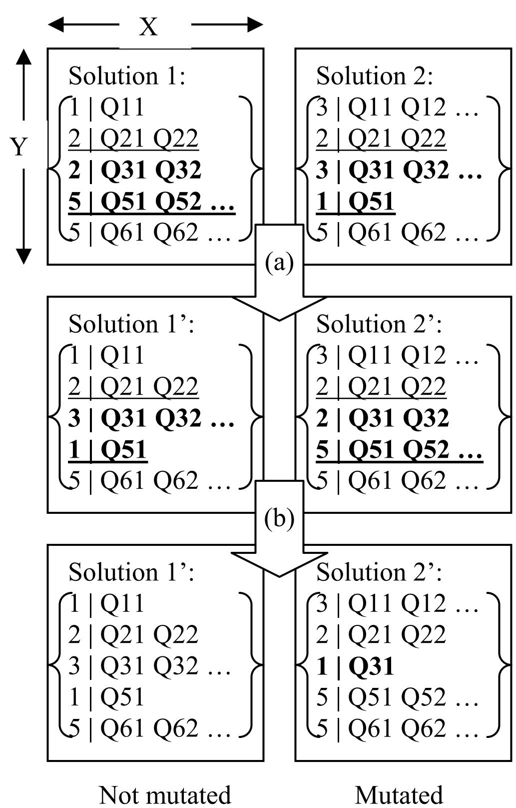

Fig. 2. The mechanism of (a) 2X and (b) mutation operation of LSGAvi

Although LSGAei and LSGAvi adopt different approaches to determine LS conditions, both of them employ the same algorithm to solve SP2: JSP after LS conditions are determined. This algorithm will be explained in the next section.

E. Job-shop Scheduling Problem

SP2 is to solve JSP after LS conditions are determined, thus SP2-GA is developed. Following SP1, a corresponding SP2 is formed after splitting n lots into n' sub-lots. A chromosome for SP2-GA is defined as the preference list of job processing priorities on each machine. For example, if m = n = 2, $L_1 = 5$, $L_2 = 8$, $S_1 = S_2 = 2$ adopting LSGAei, n' = 4 sub-lots with $Q_{11} = 2$, $Q_{12} = 3$, $Q_{21} = Q_{22} = 4$. If LSGAvi is adopted, S_i and Q_{ij} will be obtained after the algorithm terminates. Hence, a SP2 chromosome can be defined as {J_{11} J_{12} J_{21} J_{22} | J_{21} J_{11} J_{22} J_{12}}. It means that the preference list on machine 1 is $J_{11}>J_{12}>J_{21}>J_{22}$ and so on. Then Non-Delay (ND) schedule will be generated [11]. In real life manufacturing environment, the conformity of lot priorities on machines is not always applicable because of dynamic factors.

The fitness values of SP2 chromosomes are also calculated using equation (7) and will perform crossover operation according to roulette wheel selection scheme. This scheme assigns a portion to each chromosome based on its fitness on the wheel. Chromosomes with larger portions on the wheel will have more chances to be selected. The evolutionary strategy is the same as LSGAei. Job-based Order Crossover (JOX) which has been proven to preserve job order on all machines between different generations well [12], is applied to SP2-GA. Mutation operation is to swap two genes each time. Since SP2 chromosomes represent only the preference list of job processing orders, illegal offsprings will not be obtained.

IV. EXPERIMENT RESULTS

To illustrate the performance of LSGAei, the lower bound (LB) value of makespan proposed by [8] is adopted. In this connection, the LB for any m machines n lots (mxn) is obtained via equation (8). Table 2 shows LB and the value obtained by LSGAei for 3x3. For each problem, 10 experiments have been carried out. Pt_{ik} follows discrete uniform distribution [1, 10] and L_i ranges from [1, 50]. Also, setup time is ignored. Mutation rate is excluded in SP1-GA such that the results can be used to examine the sequencing ability of SP2-GA with n'. The LSGAei parameters (GEN, PS, MR) are as follows: SP1-GA (10, 10, 0.0) and SP2-GA (20, 20, 0.01).

$$LB = \max\{\sum_k Pt_{ik} \times L_i\} \quad \forall\, i \tag{8}$$

TABLE 2
COMPARISON BETWEEN LB AND RESULTS BY LSGAei

3x3	LB	LSGAei	% diff.
1	824.00	825.00	0.12
2	716.00	716.00	0.00
3	490.00	502.00	2.45
4	741.00	772.00	4.18
5	512.00	520.00	1.56
6	782.00	782.00	0.00
7	520.00	525.00	0.96
8	802.00	804.00	0.25
9	560.00	568.00	1.43
10	486.00	487.00	0.21
Avg.	643.30	650.10	1.06

From Table 2, the makespans obtained by LSGAei match very well to the theoretical LB on average. To model real problem, setup time is considered. Although the consideration of makespan leads to the reduction of the total production lead time, the benefit may be arbitrary. That's why many researchers strengthen the study of Just-In-Time (JIT) concept such that timely delivery is highlighted. An example of LS to cost control in JIT manner can be referred to [13].

Applying JIT philosophy to the current model, ec_i and tc_i are introduced. For the following experiments, SP1-GA (10,20,0.01) and SP2-GA (20,20,0.01) are implemented. mc_k varies from [1, 20]. A fixed setup time is counted during fixture changeover and pre-setup is not considered. According to [14], D_i [P(1-T-R/2), P(1-T+R/2)] where P is the sum of processing times of all operations divided by the total number of machines. Lateness factor (T) ranges from 0.1 to 0.5 and relative range of due dates (R) ranges from 0.8 to 1.8.

A. Experiment One

Consider m = n = 3, the job data of the 1st 3x3 is as follows:

$F_1 = 1$, $L_1 = 50$, $D_1 = 604$, M3 (7)→M1 (10)→M2 (4)
$F_2 = 2$, $L_2 = 9$, $D_2 = 322$, M2 (8)→M1 (10)→M3 (1)
$F_3 = 3$, $L_3 = 26$, $D_3 = 331$, M1 (9)→M3 (4)→M2 (1)
where ( ) stands for processing time per unit

In this experiment, equal size LS with idle time is studied. The penalty costs (ec_i, tc_i) of the 1st 3x3 are given as (3, 7), (4, 1) and (9, 10). Over ten 3x3 problems, three different scenarios: (i) No LS; (ii) Max. LS; (iii) With LS are examined. In general, the OPC and average deviation from due date (dev) can be minimized by splitting lots. However, splitting lots into single unit doesn't guarantee that the minimum OPC can be obtained. It is noted that for the scenario (ii) has reduced the OPC only by 20.49% and dev by 7.24% on average as compared to (i). Using LSGAei, i.e. the scenario (iii), a suitable level of splitting achieved a remarkable reduction in OPC and dev by 72.07% and 53.27% respectively as compared to (i). The results show that LSGAei is capable of deriving a suitable level of splitting with respect to the objective function.

Figure 3a shows the completion date of all sub-lots with W1 = 1 and W = 0 for the first 3x3. The final solution gives the OPC = 33657 and TSC = 1740 with $S_1 = 27$, $S_2 = 1$, $S_3 = 1$. Figure 4 shows the result with W1 = 0 and W2 = 1 for the first 3x3. The final solution gives the OPC = 218566 and TSC = 1590 with $S_1 = 1$, $S_2 = 3$, $S_3 = 1$. To compare the solutions shown in Figures 3a and 4, it is observed that LSGAei can work well with different weightings.

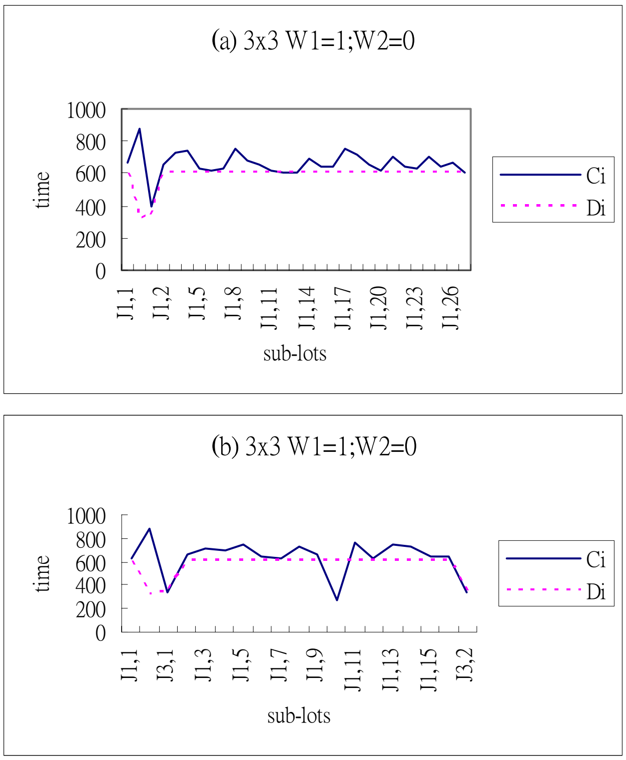

Fig. 3. Completion time of sub-lots from 1st 3x3 by (a) LSGAei and (b) LSGAvi

B. Experiment Two

Using the same setting from experiment 1, varied size LS with idle time is examined. According to [8], their algorithm, LPGA, can determine the sub-lot size for each lot with given (fixed) S_i. One shortcoming is that the result cannot tell whether it is appropriate to split a job. To deal with varied size LS, another approach is inspired by modifying the LPGA called LSGAvi. Since LPGA only works with fixed S_i, the results may not be intelligent. It is believed the selection of the split lots may improve the LS decision. To examine this point, LSGAvi is further improved in which the split lots, the split number, and the split size are all variables. Recall that the development of LSGAvi has been depicted in section III-D.

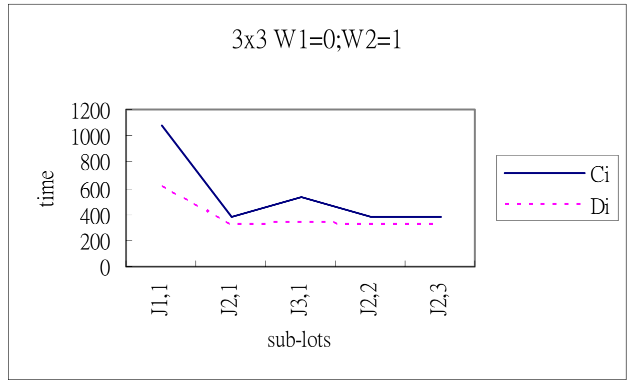

Fig. 4. Completion time of sub-lots from 1st 3x3 by LSGAei

For makespan minimization, the comparison between LSGAei and LSGAvi will be made for 3x3. Table 3 shows if LSGAvi is employed, the performance is improved. The last column indicates the difference between LB and LSGAvi. If the objective is OPC, it is observed that LSGAvi still outperforms LSGAei as shown in Table 4. Using the 1st 3x3 as example, the distribution of sub-lots completed using LSGAei and LSGAvi is plotted in Figure 3. Applying LSGAei, n' = 29 with OPC = 33657 while n' = 19 (reduced by 34.5%) with OPC = 24401 (reduced by 27.5%) by LSGAvi. Using LSGAei (Figure 3a), it is seen that sub-lots tend to be finished beyond due dates. Using LSGAvi (Figure 3b), sub-lots are finished around due dates. Since most of the split lots come from J_1 which L_1 is big (50 units), it costs less to finish it earlier as $ec_1 < tc_1$. This is one possible reason LSGAvi can obtain solutions with lower OPC.

TABLE 3
COMPARISON BETWEEN LB, LSGAei AND LSGAvi

3x3	LB	LSGAei	LSGAvi	% diff.
1	824.00	825.00	825.00	0.12
2	716.00	716.00	716.00	0.00
3	490.00	502.00	502.00	2.45
4	741.00	772.00	752.00	1.48
5	512.00	520.00	520.00	1.56
6	782.00	782.00	782.00	0.00
7	520.00	525.00	525.00	0.96
8	802.00	804.00	804.00	0.25
9	560.00	568.00	568.00	1.43
10	486.00	487.00	487.00	0.21
Avg.	643.30	650.10	648.10	0.85

TABLE 4
OVERALL PENALTY COST BY LSGAei AND LSGAvi

3x3	LSGAei (A)	LSGAvi (B)	[(A)-(B)] / (A) * 100%
1	33657.00	24401.00	27.50%
2	29645.00	21357.00	27.96%
3	71108.00	84324.00	-18.59%
4	29163.00	30143.00	-3.36%
5	28119.00	19762.00	29.72%
6	54019.00	42206.00	21.87%
7	19288.00	14614.00	24.23%
8	53149.00	31939.00	39.91%
9	22924.00	25059.00	-9.31%
10	6550.00	4745.00	27.56%
Avg.	34763.20	29855.00	-

V. ILLUSTRATIVE EXAMPLE

In this section, a workshop with a number of Computer Numerical Control (CNC) machines is chosen. Each part will go through 2 machines and Pt_{ik} [1,10]. Moreover, due to administration policies, only lots with $L_i > 10$ are allowed to be split and lots can only be split into $S_i \leq 10$ sub-lots where i = 1…n'. Also, ec_i and tc_i vary from [1, 10] for all lot i. Noted that the existing splitting policy of the company follows LSGAei.

Comparison between LSGAei and LSGAvi will be made on 5x10, 5x15 and 5x20. Table 5 shows that LSGAvi outperforms LSGAei for all test problems and the improvement increases with the problem size. This result may imply that varied size LS may help to improve the overall workshop performance. However, LSGAvi tends to increase n'. As a result, smaller sub-lots can help to offset the penalty cost induced by larger sub-lots of the same original lot. Because ec_j and tc_j are calculated per unit, small amount of early or late hours by larger sub-lots can be obtained in sacrifice of large amount of early or late hours by smaller sub-lots. For example, a lot of 30 units can be split into 6 sub-lots which sub-lot sizes are 20, 2, 2, 2, 2 and 2. Then those smaller sub-lots with 2 units can be completed in a way that the 20-unit sub-lot can be finished closer to its due date. On the other hand, LSGAei can only split lots into sub-lots of equal size. Thus, a lot with 30 units can only be split into 6 sub-lots with equal size 5 units. To this end, the offset effect cannot be applied in equal size splitting.

In the current study, a new GA approach is proposed to determine LS conditions in JSP. In general, it has been shown that varied size LS (LSGAvi) works better than equal size LS (LSGAei). Experiments have also suggested that LSGAvi performs much better if problem size grows. More importantly, LSGAvi can solve LS problem by determining the LS conditions and the sub-lot processing order simultaneously. According to authors' knowledge, there is no similar approach to determine LS conditions in JSP. Although the proposed model works well to the objectives, an intelligent method will be studied to obtain the optimal weightings of the objective function.

TABLE 5
COMPARISON BETWEEN LSGAei AND LSGAvi ON TESTING PROBLEMS

Problems	LSGAei (A)	LSGAvi (B)	[(A)-(B)] / (A) * 100%
5x10	295653.00	275668.00	6.76%
5x15	214203.00	177698.00	17.04%
5x20	456283.00	376998.00	17.38%

REFERENCES

[1] Woo S., Kang S. and Park J., 1997, A batch splitting heuristic for dynamic job shop scheduling problem. Computers and Industrial Engineering, 33, 781-784.
[2] Jeong H. I., Park J. and Leachman R. C., 1999, A batch splitting method for a job shop scheduling problem in an MRP environment. International Journal of Production Research, 37, 3583-3598.
[3] Kalir A. A. and Sarin S. C., 2001, A near-optimal heuristic for the sequencing problem in multiple-batch flow-shops with small equal sublots. International Journal of Management Science, 29, 577-584.
[4] Kumar S., Bagchi T. P. and Sriskandarajah C., 2000, Lot streaming and scheduling heuristics for m-machine no-wait flowshops. Computers and Industrial Engineering, 38, 149-172.
[5] Yoon S. H. and Ventura J. A., 2002, An application of genetic algorithms to lot-streaming flow shop scheduling. IIE Transactions, 34, 779-787.
[6] Şen A. and Benli Ö. S., 1999, Lot streaming in open shops. Operations Research Letters, 23, 135-142.
[7] Trietsch D. and Baker K. R., 1993, Basic techniques for lot streaming. Operations Research, 6, 1065-1076.
[8] Dauzère-pérès S. and Lasserre J. B., 1997, Lot streaming in job-shop scheduling. Operations Research, 45, 584-595.
[9] Holland J. H., 1975, Adaption in natural and artificial systems. The University of Michigan Press, Ann Arbor.
[10] Cheng R., Gen M. and Tsujimura Y., 1999, A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies. Computers & Industrial Engineering, 36, 343-364.
[11] Croce F. D., Tadei R. and Volta G., 1995, A genetic algorithm for the job shop problem. Computers and Operations Research, 22, 15-24.
[12] Ono I., Yamamura M. and Kobayashi S., 1996, A genetic algorithm for job-shop scheduling problems using job-based order crossover. Proceedings of International Conference on Evolutionary Computation 1996, 547-552.
[13] Kim S. L. and Ha D., 2003, A JIT lot-splitting model for supply chain management: enhancing buyer-supplier linkage. International Journal of Production Economics, 86, 1-10.
[14] Potts C. N. and Van Wassenhove L. N., 1982, A decomposition algorithm for the single machine total tardiness problem. Operations Research Letters, 1, 177-181.
[15] Potts C. N. and Baker K. R., 1989, Flow shop scheduling with lot streaming. Operations Research Letters, 8, 297-303.