Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

The Application of Lot Streaming to Assembly Job Shop under Resource Constraints

F. T. S. Chan*#, T. C. Wong* and L. Y. Chan*

**Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong Hong Kong (#e-mail: ftschan@ hkucc.hku.hk).*

Abstract: Assembly job shop problem (AJSP) is an extension of classical job shop problem (JSP). AJSP first starts with a JSP and appends an assembly stage after job completion. Lot Streaming (LS) technique is defined as the process of splitting lots into sub-lots such that successive operation can be overlapped. In this paper, the previous study of LS to AJSP will be extended by introducing resource constraints. To reduce the computational effort, we propose a new Genetic Algorithm (GA) approach which is the modification of the algorithm in our previous paper. A number of test problems are conducted to examine the performance of the new GA approach. Moreover, the single GA approach will be compared with a single Particle Swarm Optimization (PSO) approach. Computational results suggest that the new algorithm can outperform the previous one and the PSO approach with respect to the objective function.

1. INTRODUCTION

For classical Job Shop Problem (JSP), there are *m* machines and *n* jobs. Each machine represents one operation. A job or lot which is defined as a batch of identical items should be processed on all machines until all of its operations are completed. The general assumptions are that each machine can process one job at most and each job can only visit each machine once. The processing sequence of jobs should be strictly followed. Lot Streaming (LS) or job splitting technique depicts a process of splitting lots into sub-lots such that successive operations of the same lot can be processed in parallel on different machines. In this paper, the application of LS to Assembly Job Shop Problem (AJSP) which is an extension of JSP will be examined. AJSP first starts with JSP and then appends an assembly stage after job completion. It means that the completed jobs which belong to the Bill-Of-Material (BOM) of the same product can be assembled. The product assembly can start once all jobs of the same BOM are completed at the JSP stage. To be realistic, part sharing is allowed such that completed jobs from distinct BOMs may also be assembled. In this connection, two types of jobs have been defined as: Unique and Standard. Unique job type is specific to only one product and only standard job type can be shared among distinct products. For example, suppose the BOM of Product 1 or P1 contains Job 1 or J1 (unique type) and J2 (standard type) while the BOM of P2 includes J3 (unique type) and J4 (standard type). Since both J2 and J4 are of standard job type, it means that J2 can substitute J4 for the assembly of P2 or J4 can replace J2 for the assembly of P1. One main modification to the previous study (Chan *et al.*, 2007b) is that resource constraints are introduced. The renewable resources defined are Fixture and Tool. Details will be given in Section 3.2. To reduce the computational effort, a new Genetic Algorithm (GA) approach will be developed to solve this problem with modified crossover and mutation operators.

The paper is organized as follows. The literature review will be discussed in the next section. In Section 3, the problem background and formulations will be introduced. In Section 4, the proposed approach will be depicted. Computation results will be reported in Section 5. Discussions will be presented in Section 6 together with future research areas.

2. LITERATURE REVIEW

Being firstly introduced by Reiter (1966), LS technique is a methodology to split a job into smaller sub-jobs such that successive operations of the same job can be processed in parallel. Thus, the lead time of the whole job can be possibly shortened. Prior to job splitting, the nature of job size and the sub-job type should be defined. In general, the job size can be defined as discrete or continuous. Discrete job size means a job contains an integer number of identical items. Continuous job size can be a real number. Also, the sub-lot type can be defined as variable or consistent. Variable type means that the sub-job size may vary between successive machines. Consistent type restricts that the sub-job size is fixed. Over the past few years, LS has been prevalently applied to Flow Shop Problem (FSP) (Chen and Steiner, 2003; Kalir and Sarin, 2001a, b; Kumar *et al.*, 2000; Liu, 2006; Marimuthu *et al.*, 2006; Martin, 2006; Smunt *et al.*, 1996; Yoon and Ventura, 2002a, b) which only allows one route for all jobs. In fact, this "one-route-for-all" feature has enabled LS to work its very best in FSP. In contrast, LS seems not very promising in JSP and Open Shop Problem (OSP). Even so, some studies about LS to JSP (Chan *et al.*, 2007a; Dauzère-pérès and Lasserre, 1997; Jeong *et al.*, 1999; Smunt *et al.*, 1996) and OSP (Şen and Benli, 1999) can be found. According to Trietsch and Baker (1993), LS approach can be classified into 4 types, but only two types are studied, i.e. type II: Equal size sub-jobs with intermittent idling that allows idle time between sub-jobs on the same machine, and type IV: Varied size sub-jobs with intermittent idling. For a comprehensive review on LS, please refer to Chang and Chiu (2005).

Assembly is usually defined as the process to construct a final product from its components. The complexity of a product mainly depends on the number of its components and the assembly levels. A typical product structure with 4 assembly levels is presented in Figure 1. The top level is Product 1 (P1). The second level contains Assembly 1 (A1), Assembly 2 (A2) and Component 1 (C1). A1 is the assembly of A3 and C2. C4 and C5 are assembled for A3 and so on. To append assembly stage after JSP, the problem then becomes AJSP. Recently, many researches have been dedicated to AJSP (Gravel *et al.*, 2000; Guide *et al.*, 2000; Guo *et al.*, 2006; Mckoy and Egbelu, 1998; Mckoy and Egbelu, 1999). Guide *et al.* (2000) have discussed about the priority scheduling polices, i.e. dispatching rules, in repair shop with no spares. In fact, the repair shop is the same as AJSP with respect to our definition. They further classify a 3-level component matching as: serial number specific, common and the mix of them. Serial number specific means that each component is unique to one product only. Common level allows all components to be shared among all products. The last level is the mix of the previous two levels. In this paper, we introduce a 4-level part sharing ratio such that all 3 levels of component matching can be considered and controlled. However, lot splitting is ignored. Guo *et al.* (2006) have developed a universal mathematical model with genetic optimization process to an industrial case study. Significant improvements have been observed but lot splitting again is not considered. McKoy and Egbelu (1998) have presented a 12-step heuristic to minimize the production flow time for AJSP. Comparisons are made between their proposed heuristic and the mixed integer linear program (MILP) on some test problems. The results have suggested MILP performs better in terms of solution quality. However, MILP requires substantial computational time to obtain the optimality. However, LS is clearly not considered.

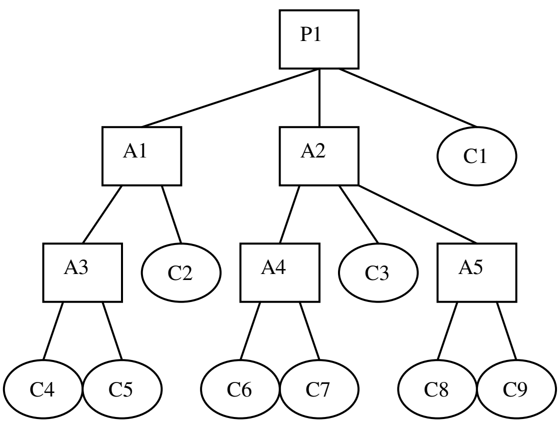

Fig. 1. A typical product structure

3. PROBLEM BACKGROUND

3.1 Notations

p	Total type of products
m	Total number of machines
n	Total number of lots
n*	Total number of sub-lots
P_h	Product h
DM_h	Demand of product h
DD_h	Due date of product h
CN_h	Total number of components of product h
At_h	Assembly time of product h
Ct_h	Delivery time of product h
M_i	Machine i
CF_i	Current fixture type on M_i
L_{hj}	Lot j of product h
Q_{hj}	Lot size of L_{hj}
F_{hj}	Fixture type of L_{hj}
T_{hj}	Tool type of kth operation of L_{hj}
MS_{hjk}	Machine for kth operation of L_{hj}
Pt_{hjk}	Processing time of kth operation of L_{hj}
SU_{hjk}	Setup time of kth operation of L_{hj}
St_{hjk}	Start time of kth operation of L_{hj}
Ct_{hjk}	Completion time of kth operation of L_{hj}
SN_{hj}	Sub-lot number of L_{hj}
L_{hjs}	sth sub-lot of L_{hj}
Q_{hjs}	Lot size of L_{hjs}
St_{hjsk}	Start time of kth operation of L_{hjs}
Ct_{hjsk}	Completion time of kth operation of L_{hjs}
mc_i	Machining cost of M_i per hour
lc_h	Late cost of P_h per unit per hour

3.2 Problem Formulations

In this paper, AJSP contains a Work Station, an Inventory Station and an Assembly Station. The Work Station contains *m* distinct machines. For each planning period, there are *p* product types. Each product P_h contains $CN_h \in [2,10]$ distinct components $\forall\, h = 1...p$, i.e. its BOM. The demand of P_h is given as $DM_h \in [1,50]$ unit(s). According to Potts and Van Wassenhove (1982), the due date of P_h can be generated from a discrete uniform distribution $DD_h \in [\gamma\bullet(1-\alpha-\beta/2), \gamma\bullet(1-\alpha+\beta/2)]$ hour(s) where γ as defined by (2), $\alpha \in [0.1, 0.5]$, and $\beta \in [0.8, 1.8]$ respectively. The assembly time of P_h at the Assembly Station is $At_h \in [10, 50]$ hour(s). Aforementioned, a lot is composed of identical components only, thus there are total *n* lots as shown in (3) and the lot size of L_{hj} is Q_{hj} by (4). The due date of L_{hj} is the same as P_h. Similar to JSP, L_{hj} is required to be processed on *m* machines with processing time $Pt_{hjk} \in [1,10]$ hour(s) and setup time $SU_{hjk} \in [1,20]$ hour(s) $\forall\, k = 1...m$. L_{hj} can be processed on M_i once and SU_{hjk} is counted only if $CF_i \neq F_{hj}$, otherwise $SU_{hjk} = 0$. To simulate various system starting conditions, congestion index (CI) is introduced in 4 levels: 0, 0.25, 0.5 and 0.75. If CI = 0, it means the system is empty, i.e. no congestion. If CI = 0.75, it refers to the earliest starting time on each machine is at least 0.75 times the total processing time required, i.e. highly congested. For each P_h, there are at most 5 different standard types of lots. The job type is controlled by a part sharing ratio (R). In this study, R has 4 levels: 0%, 30%, 50% and 70%. For example, if R is 30%, each job of product 1 has 30% chances to become a standard job type or 70% to become a unique job type. It is assumed that each product can only have at most 5 jobs of 5 different standard job types. Unlike the previous study (Chan *et al.*, 2007b), resource constraints are considered. Hence, based on a typical CNC machining workshop, two resource units are introduced as: Fixture and Tool. Fixture is used to mount the components on the machines for processing and it is unique to each component. If $n = 10$, there are 10 distinct fixture type. In this connection, only the sub-lots of the same lot may share the same fixture. Likewise, each machine should equip tools to perform operations on the components. Each operation of the components should be assigned one tool and there is no repeated tool for the same component. If there are 5 operations, then there are 5 distinct tools for the same component. However, some operations of other components may share the same tool. In order to successfully perform an operation on a component, the machine should equip the associated fixture and tool. Otherwise, no operation can start on the machine, i.e. resource shortage. Since resources are defined as renewable, it means that each fixture and tool can be instantly available once their previous operation has been completed. In this study, the total number of fixture types equals to n and the total number of tool types is (m*n*0.25). If resources are not limited, each resource has m copies, i.e. one copy for each machine (high resource level). In contrast, each resource only has one copy, i.e. m machines share one copy (low resource level). To examine the impact of resource limitation on LS to AJSP, these two resource levels will be studied and the results will be reported in Section 5.

L_{hj} can only be processed in the Work Station and all finished lots should be stored at the Inventory Station until at least one unit of P_h can be assembled. Then, L_{hj} will be transferred to the Assembly Station for product assembly and P_h can be delivered to the customer. Equation (4) defines the original batch size of each L_{hj}. The delivery time of P_h is Ct_h as shown in (5). It means that P_h can be delivered only if all L_{hj} are completed. This equation is valid if LS is not allowed. However, if LS is permitted, (5) is not applicable. In this paper, only types II and IV LS models which allow idle time between sub-lots on the same machine are considered with discrete lot size and consistent sub-lot type. With LS, L_{hj} can be split into SN_{hj} sub-lots. The total number of L_{hjs} (n^*) is obtained from (6). The due date, fixture type, job type, tool type, machine sequence, processing time, and setup time of sub-lot L_{hjs} are the same as that of the original lot L_{hj}. Equation (7) restricts that the total lot size of sub-lots should satisfy the original lot size. The late cost of P_h per unit per hour is $lc_h \in [0.1, 1.0]$. It is noted that the late cost is calculated per product. The research objective is to minimize the Lateness Cost. If LS is not allowed, the objective function (Z) is defined by (1). With LS, Z is considered with respect to time as there may be different Ct_h values if L_{hj} can be split, i.e. Z(t).

Objective:

$$Min.\ Z = \sum_{h=1}^{p} \{Ct_h - DD_h\}^{+} \bullet DM_h \bullet lc_h \tag{1}$$

Parameter Constraints:

$$\gamma = (\sum_{h=1}^{p} \sum_{j=1}^{CN_h} \sum_{k=1}^{m} Pt_{hjk} \bullet Q_{hj}) / m \tag{2}$$

$$n = \sum_{h=1}^{p} CN_h \tag{3}$$

$$Q_{hj} = DM_h \quad \forall\ h = 1...p,\ j = 1...CN_h \tag{4}$$

$$Ct_h = \max.\left(Ct_{hjm}\right) + At_h \ \ where\ k = m \quad \forall\ h = 1...p,\ j = 1...CN_h \tag{5}$$

$$n^* = \sum_{h=1}^{p} \sum_{j=1}^{CN_h} SN_{hj} \tag{6}$$

$$\sum_{s=1}^{SN_{hj}} Q_{hjs} - Q_{hj} = 0 \quad \forall\ h = 1...p,\ j = 1...CN_h \tag{7}$$

Same as JSP, lots are processed on m distinct machines. If LS is not considered, (8) restricts that all lots should be processed with respect to the predefined processing sequence. Also, each machine can process only one lot and pre-emption is prohibited. The queuing time of lots between kth and (k+1)th operation is defined in (9). Furthermore, all lots are ready for processing at the beginning of each planning period defined by (10) unless CI is counted. Likewise, the same set of operational constraints (8) - (10) is also applicable to all sub-lots once LS is allowed by substituting St_{hjk} with St_{hjsk}, $St_{hj(k+1)}$ with $St_{hjs(k+1)}$, Q_{hj} with Q_{hjs}, Ct_{hjk} with Ct_{hjsk}, St_{hj0} with St_{hjs0} and Ct_{hj0} with Ct_{hjs0} for s = 1…SN_{hj}.

Operational Constraints:

$$St_{hj(k+1)} \geq St_{hjk} + Pt_{hjk} \bullet Q_{hj} + SU_{hjk} \bullet \mu_{hjk} \quad \forall\ h = 1...p,\ j = 1...CN_h, k = 1...m \tag{8}$$

where $\mu_{hjk} = \begin{cases} 0\ if\ F_{hj} = CF_{MS_{hjk}} \\ 1 if\ F_{hj} \neq CF_{MS_{hjk}} \end{cases}$

$$St_{hj(k+1)} - Ct_{hjk} \geq 0 \quad \forall\, h = 1...p,\, j = 1...CN_h,\, k = 1...m \tag{9}$$

$$St_{hj0} = Ct_{hj0} = 0 \quad \forall\, h = 1...p,\, j = 1...CN_h \tag{10}$$

4. THE PROPOSED ALGORITHM

With respect to our previous study (Chan *et al.*, 2007b), if LS is allowed, the current problem can be divided into 2 sub-problems, Sub-Problem 1 (SP1): Determination of sub-lot combinations and Sub-Problem 2 (SP2): AJSP with all sub-lots. To be effective and efficient, one of the robust evolutionary algorithms, GA, is proposed to solve SP1 and SP2. In fact, SP1 is already a complex problem, not mention about SP2 which is NP-hard. For the old approach, a GA is developed to solve SP1 and SP2 apiece, i.e. two GAs in total. One shortcoming is that the computational time is significant due to iterative computation between two GAs. In this connection, we propose a new approach in which a single GA is developed to solve SP1 and SP2 simultaneously. Figure 2 shows the mechanism of the proposed algorithm. The modification details will be given in Section 4.3. Hereafter, the previous algorithm (two GAs) is denoted as PAL and the new algorithm (single GA) is called NAL.

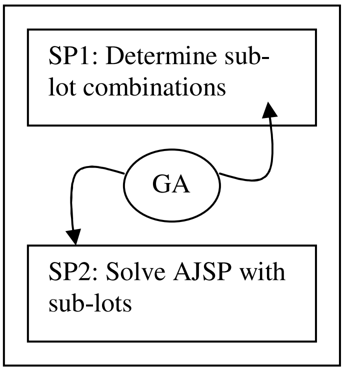

Fig. 2. The proposed algorithm

4.1 Genetic Algorithm

GA which was first introduced by Holland (1975) has been widely applied for solving complex combinatorial problems. Its principles follow the natural evolution and the rule of Survival of the Fittest. It means that good solutions will have greater chances to survive and mate with others. Elitism which means the best chromosome can always be survived is applied. The GA mechanism executes from pool to pool until terminating criteria are met. Roulette Wheel Selection Scheme is implemented to select chromosomes for crossover operation. Mutation is essential to prevent premature convergence. It is noted that the number of chromosomes in a pool is referred to population size (PS) and the total number of pools is regarded as generation number (GEN). The transformation of objective value to fitness value is governed by (14). The objective value (Z) is obtained from (1).

$$FV_c = (MAX - Z_c + MIN)/AVG \quad \forall\, c = 1...PS \tag{14}$$

FV_c: Fitness value of cth chromosome
MAX: The maximum objective value of the same generation
Z_c: Objective value of cth chromosome
MIN: The minimum objective value of the same generation
AVG: The average objective value of the same generation

4.2 Particle Swarm Optimization

PSO was first introduced by Eberhart and Kennedy (1995). Its principle is based on the behaviour of flying birds and their means of information exchange. It combines local search and global search leading to effective searching ability. For recent years, PSO is one of the common evolutionary algorithms to JSP, FSP, OSP and assembly-related problems (e.g. Allahverdi and Al-Anzi, 2006; Al-Anzi and Allahverdi, 2007). In this study, the mechanism of the PSO which was defined by Allahverdi and Al-Anzi (2006) is implemented and it will be compared with the new GA approach to AJSP with LS. The computational results will be also reported in Section 5.

4.3 The Modification to the Previous Algorithm

The main difference between the old and new approach is that the previous GA approach works with two GAs while the new GA approach works with a single GA to SP1 and SP2. For the old method, two GAs should be executed iteratively so as to obtain the best solution. Hence, the computational effort would be significant if the problem size becomes huge. In this connection, we propose a new approach such that one GA is capable of solving SP1 and SP2 simultaneously. As a single GA is considered, the chromosome structure should be modified as shown in Figure 3. In X-Z dimension, the chromosome is the same as the one to SP1 by the previous approach. In X-Y dimension, the chromosome is also the same as the one to SP2 by the previous approach. For the first time, we combine two chromosomes into a single one as a 3D structure. Hence, this new 3D chromosome can present a complete solution to the research problem, i.e. the lot splitting conditions and the sub-lot processing sequence on machines.

In X-Z dimension, (X, Z=1) = SN_X is the sub-lot number of job X for X = 1…*n*. (X, Z) = Q_{XZ*} is the sub-lot size for X = 1…*n* and Z = 2..SN_X+1. Noted that Z* = (Z-1). In X-Y dimension, (X, Y) is the processing preference on machines. For example, if Job 1 is located at (2, Y) and Job 2 is at (4, Y), then it means that Job 1 is more preferable than Job 2 on machine Y for Y = 1 …*m*. The direction X here refers to the processing preference for *n* jobs. It is noted that the sub-lots are scheduled as close as its original lot, hence the length of direction X remains *n* but not *n**. To successfully perform crossover operation, JOX which was defined by Ono *et al.* (1996) is implemented. Suppose we have two chromosomes, C1 and C2. Figure 4a illustrates how JOX can work with C1 and C2. From the figure, Job 2 and Job 3 are preserved. Hence, the LS conditions and the processing preferences of Job 2 and Job 3 are also preserved. Then the non-preserved genes are interchanged between C1 and C2. Figure 4b shows the proposed mutation operation. From the figure, mutation has been applied to C1 only. First, the mutation mechanism will randomly select one job out of *n* jobs. Then the sub-lot number and the sub-lot size of the chosen job will be re-generated in X-Z dimension. For this case, Job 1 is chosen. Second, another job will be randomly selected. In this case, Job 3 is selected. Then Job 1 and Job 3 will be swapped on a machine in X-Y dimension. According to authors' knowledge, there is no similar GA-based approach to AJSP with LS.

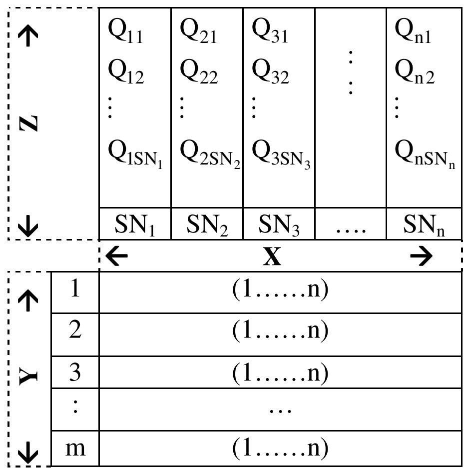

Fig. 3. The proposed chromosome structure

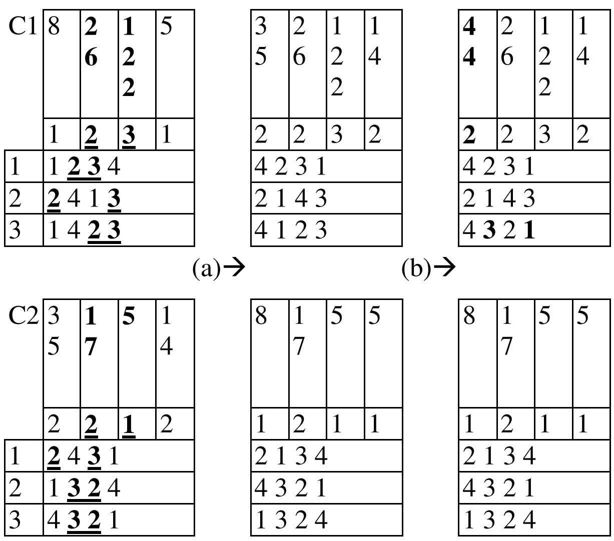

Fig. 4. (a) JOX and (b) Mutation of the new GA approach

5. COMPUTATIONAL RESULTS

In this section, PAL and NAL will be compared. Then the NAL or the new GA approach will be compared with PSO under different experimental settings as shown in Table 1. Form the table, RL = Resource Level, R = Part Sharing Ratio, CI = Congestion Index and Runs = number of runs for each setting. Hence, we have total (2*4*4*10) = 320 experiments for each test problem. Due to the page limit, the results for m = {3} and p = {3, 5, 7, 10} will be reported only. The problem is identified as mxp: 3x3, 3x5, 3x7 and 3x10. In this experiment, crossover rate = 0.8, mutation rate = 0.01, PS = 40 and GEN = 100 after some preliminary tests. The same number of iterations is also applied to PSO, i.e. 40*100 = 4000.

Table 1. Experimental settings

Parameters	
RL	High, Low
R	0, 30, 50, 70
CI	0, 0.25, 0.5, 0.75
Runs	10

First, PAL and NAL are compared on a number of test problems. It is observed that the optimization results are comparable. However, NAL can outperform PAL in terms of computational effort by at least 50%. Since NAL takes less computational effort, it will be compared with another well-known evolutionary algorithm, PSO. With respect to the previous study (Chan *et al.*, 2007b), three LS modes are examined: No LS (No), Equal Sized LS (ES) and Varying Size LS (VS). It is expected that ES can outperform No and VS. Preliminarily, we assume GA works better than PSO. Hence, we set ES with GA (GA-ES) as datum and compare it with others in percentage. Therefore, if positive percentage difference is obtained, it means GA-ES is better. For 3x3 with high RL, GA-ES outperforms GA-No by 9.28% to 24.12% and GA-VS by 1.52% to 10.7%. It reinforces that ES is the best LS mode. If PSO is studied, PSO-ES is also the best as compared to PSO-No and PSO-VS. However, GA-ES can overpower PSO-ES by 0.01% to 6.15% except the case with R = 70% and CI = 0.25. For 3x5 with high RL, GA-ES is still best option. It outranks GA-No by 4.24% to 20.24% and GA-VS by 0.76% to 15.62%. Also, GA-ES overwhelms PSO-ES by 4.18% to 28.67% for all cases. For 3x7 with high RL, GA-ES outperforms GA-No by 0.97% to 33.1% except R = 0% and CI = 0.5. GA-ES overpowers GA-VS by 0.42% to 36.67% in 13 out of 16 cases. Likewise, PSO-ES is the best option in 11 out of 16 cases. Nevertheless, ES is still the best recommended LS mode. Moreover, GA-ES works better than PSO-ES by 3.57% to 39.02% for all cases. For 3x10 with high RL, GA-ES outperforms GA-No by 0.58% to 43.21% in 13 out of 16 cases and overwhelms GA-VS by 2.31% to 13.86% in 10 out of 16 cases. Also, PSO-ES is the best in 10 out of 16 cases. To compare GA and PSO, GA-ES still overwhelms PSO-ES by 6.13% to 39.7% except R = 50% and CI = 0. For all test problems with low RL, the LS effect diminishes, i.e. the percentage differences between ES mode and the rest (No LS and VS) become smaller. Nevertheless, LS still can work well under resource constraints. In terms of average computational time, the minimum is 10 seconds for 3x3 and the maximum is about 636 seconds or 10 minutes for 3x10.

6. DISCUSSION

From the computational results, there are several observations: (1) ES mode is the best recommended approach to AJSP with LS using GA and PSO, (2) LS effect may diminish if the ratio of *m* to *p* becomes small, (3) LS effect may diminish if the system is highly congested, (4) Part sharing has no obvious conflict with LS, (5) Resource constraints may diminish LS effect, and (6) Last but not least, GA works better than PSO for most of the test problems with less computational effort. In general, the potential of LS to AJSP has not yet been fully quantified and studied. In this connection, bigger test problems will be examined such as m = 5 and 7.

REFERENCES

Allahverdi, A. and Al-Anzi, F.S. (2006). A PSO and a tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application. *Computers & Operations Research*, **33**, 1056-1080.

Al-Anzi, F.S. and Allahverdi, A. (2007). A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times. *European Journal of Opoerational Research*, **182**, 80-94.

Chan, F.T.S., Wong, T.C. and Chan, L.Y. (2007a). Lot streaming for product assembly in job shop environment. *International Journal of Robotics and Computer Integrated Manufacturing*. (In Press)

Chan, F.T.S., Wong, T.C. and Chan, L.Y. (2007b). The application of lot streaming to assembly job shop with part sharing. *The 8th APIEMS conference*, Taiwan (Accepted for publication)

Chang, J. H. and Chiu, H. N. (2005). A comprehensive review of lot streaming. *International Journal of Production Research*, **43**, 8, 1515-1536.

Chen, J. and Steiner, G. (2003). On discrete lot streaming in no-wait flow shops. *IIE Transactions*, **35**, 91-101.

Dauzère-pérès, S. and Lasserre, J. B. (1997). Lot streaming in job-shop scheduling. *Operations Research*, **45**, 4, 584-595.

Eberhart, R. C. and Kennedy, J. (1995). A new optimizer using particle swarm theory. *Proceedings of the 6th international symposium on micro machine and human science*. IEEE service center. Piscataway, NJ, Nagoya, Japan, 39-43.

Gravel, M., Price, W. and Gagne, C. (2000). An interactive tool for designing manufacturing cells for an assembly job-shop. *International Journal of Production Research*, **38**, 2, 309-322.

Guide Jr., V. D. R., Srivastava, R. And Kraus, M. E. (2000). Priority scheduling polices for repair shops. *International Journal of Production Research*, **38**, 4, 929-950.

Guo, Z. X., Wong, W. K., Leung, S. Y. S., Fan, J. T. and Chan. S. F. (2006). Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: a case study based on the apparel industry. *Computers & Industrial Engineering* (In Press)

Holland, J. H. (1975). Adaption in natural and artificial systems. The University of Michigan Press, Ann Arbor.

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**, 15, 3583-3598.

Kalir, A. A. and Sarin, S. C. (2001a). Optimal solutions for the single batch, flow shop, lot-streaming problem with equal sublots. *Decision Sciences*, **32**, 2, 387-397.

Kalir, A. A. and Sarin, S. C. (2001b). A near-optimal heuristic for the sequencing problem in multiple-batch flow-shops with small equal sublots. *Omega*, **29**, 577-584.

Kumar, S., Bagchi, T. P. and Sriskandarajah, C. (2000). Lot streaming and scheduling heuristics for m-machine no-wait flowshops. *Computers & Industrial Engineering*, **38**, 149-172.

Liu, J. (2006). Single-job lot streaming in m-1 two-stage hybrid flowshops. *European Journal of Operational Research* (In Press)

Marimuthu, S., Ponnambalam, S. G. and Jawahar, N. (2006). Evolutionary algorithms for scheduling m-machine flow shop with lot streaming. *International Journal of Robotics and Computer Integrated Manufacturing* (In Press)

Martin, C. H. (2006) A hybrid genetic algorithm / mathematical programming approach to the multi-family flowshop scheduling problem with lot streaming. *Omega* (In Press)

Mckoy, D. H. C. and Egbelu, P. J. (1998) Minimizing production flow time in a process and assembly job shop. *International Journal of Production Research*, **36**, 8, 2315-2332.

Mckoy, D. H. C. and Egbelu, P. J. (1999). Production scheduling in a process and assembly job shop. *Production Planning & Control*, **10**, 1, 76-86.

Ono, I., Yamamura, M. and Kobayashi, S. (1996). A genetic algorithm for job-shop scheduling problems using job-based order crossover, *Proceedings of an International Conference on Evolutionary Computation*, Nayoya University, Japan, 547-552.

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.

Reiter, S. (1966). A system for managing job-shop production. *Journal of Business*, **39**, 3, 371-393.

Şen, A. and Benli, Ö. S. (1999) Lot streaming in open shops. *Operations Research Letters*, **23**, 135-142.

Smunt, T. L., Buss, A. H. and Kropp, D. H. (1996). Lot splitting in stochastic flow shop and job shop environment. *Decision Sciences*, **27**, 2, 215-238.

Trietsch, D. and Baker, K. R. (1993). Basic techniques for lot streaming. *Operations Research*, **41**, 6, 1065-1076.

Yoon, S. H. and Ventura, J. A. (2002a). An application of genetic algorithms to lot-streaming flow shop scheduling. *IIE Transactions*, **34**, 779-787.

Yoon, S. H. and Ventura, J. A. (2002b). Minimizing the mean weighted absolute deviation from due dates in lot-streaming flow shop scheduling. *Computers & Operations Research*, **29**, 1301-1315.