

# No-Wait Lot-Streaming Flow Shop Scheduling

Suk-Hun Yoon<sup>†</sup>

Department of Industrial and Information Systems Engineering, Soongsil University, Seoul, 156-743

## 비정체 로트 – 스트리밍 흐름공정 일정계획

윤석훈

승실대학교 산업정보시스템공학과

Lot-streaming is the process of splitting a job (lot) into a number of smaller sublots to allow the overlapping of operations between successive machines in a multi-stage production system. A new genetic algorithm (NGA) is proposed for minimizing the mean weighted absolute deviation of job completion times from due dates when jobs are scheduled in a no-wait lot-streaming flow shop. In a no-wait flow shop, each subplot must be processed continuously from its start in the first machine to its completion in the last machine without any interruption on machines and without any waiting in between the machines. NGA replaces selection and mating operators of genetic algorithms (GAs), which often lead to premature convergence, by new operators (marriage and pregnancy operators) and adopts the idea of inter-chromosomal dominance. The performance of NGA is compared with that of GA and the results of computational experiments show that NGA works well for this type of problem.

**Keywords:** scheduling, no-wait flow shop, lot-streaming, mean weighted absolute deviation, genetic algorithms

## 1. Introduction

Lot-streaming is the process of splitting a job (lot) into a number of smaller sublots so that successive operations can be overlapped in a multi-stage production system. This process is illustrated by the two-job, equal subplot, three-machine, no-wait lot-streaming flow shop shown in Figure 1, where jobs 1 and 2 are divided into three and two sublots, respectively. The processing times of jobs 1 and 2 are 3 and 2 time units on machine 1, 3 and 6 time units on machine 2, and 6 and 4 time units on machine 3, respectively. The due dates of jobs 1 and 2 are in 8

and 14 time units. If the jobs are not split into sublots, the completion times of jobs 1 and 2 will be 12 and 16 time units, and both jobs will have a delay of 4 time units (schedule 1). As <Figure 1(b)> shows, when the jobs are split into sublots, the completion times of jobs are reduced to 8 and 14 time units, respectively, and both jobs can be delivered on time (schedule 2).

The potential benefits of lot-streaming in batch manufacturing are pointed out in Jacobs and Bragg (1988) and Benjaafar (1996). These benefits include reduction of production lead times, reduction of work-in-process (WIP) inventory and associated WIP costs, reduction of interim storage and space

---

This work was supported by the Soongsil University Research Fund.

<sup>†</sup> Corresponding author : Suk-Hun Yoon, Department of Industrial and Information Systems Engineering, Soongsil University, Seoul, 156-743,  
Fax : 82-2-825-1094, E-mail : yoon@ssu.ac.kr

Received May 2004, accepted May 2004 after 1 revisions.

requirements, reduction of material handling system capacity requirements, and improvement of product delivery. An extensive review of research in this area is presented in Potts and Wassenhove (1992) and Graves and Kostreva (1986). Sriskandarajah and Wagneur (1999) have shown that the multiple-job, two-machine, no-wait lot-streaming flow shop scheduling problem to minimize the makespan could be solved using a modified Gilmore and Gomory's algorithm (Pinedo, 2002).

In many practical situations, frequently encountered in chemicals processing and petro-chemical production environments, and hot metal rolling industries, where the metals have to be processed continuously at high temperature, delays between operations are prohibited (Gershwin, 1994). This paper presents a solution methodology for an  $n$ -job,  $m$ -machine, no-wait flow shop scheduling problem with equal-size sublots in which the objective is to minimize the mean weighted absolute deviation from due dates. With the increasing interest in just-in-time (JIT) production systems, the trend of research has changed to scheduling problems with earliness and tardiness costs. In a JIT production system, it is highly desirable that jobs be completed exactly on their due dates. Since most scheduling problems involving earliness and tardiness penalties are not tractable, it is usually very difficult to find a

global optimum solution by general local search algorithms such as the adjacent pairwise interchange method. Recently, several methods have been proposed to escape from local optima and search for global or near-optimal solutions. Among these, meta-heuristic methods such as genetic algorithms, simulated annealing, and tabu search, have been successful in solving combinatorial optimization problems (Reeves, 1993).

GAs have been applied to various problems that could not have been readily solved with conventional computational techniques. In spite of their desirable properties, basic GAs can still fail for a variety of reasons, including choice of representation that is not consistent with the crossover operator, failure to represent problem specific information, and convergence to local optima (premature convergence). In Section 2, the  $n$ -job,  $m$ -machine, equal-size subplot, no-wait lot-streaming flow shop problem is defined. For a given job sequence (individual), a linear programming (LP) formulation is presented to obtain optimal subplot starting and completion times, since the objective function value of a sequence may be improved by inserting idle times between sublots and between jobs. The objective function values of the LP are transformed to obtain fitness values of individuals. An NGA is developed to overcome the premature convergence of GAs in Section 3. NGA

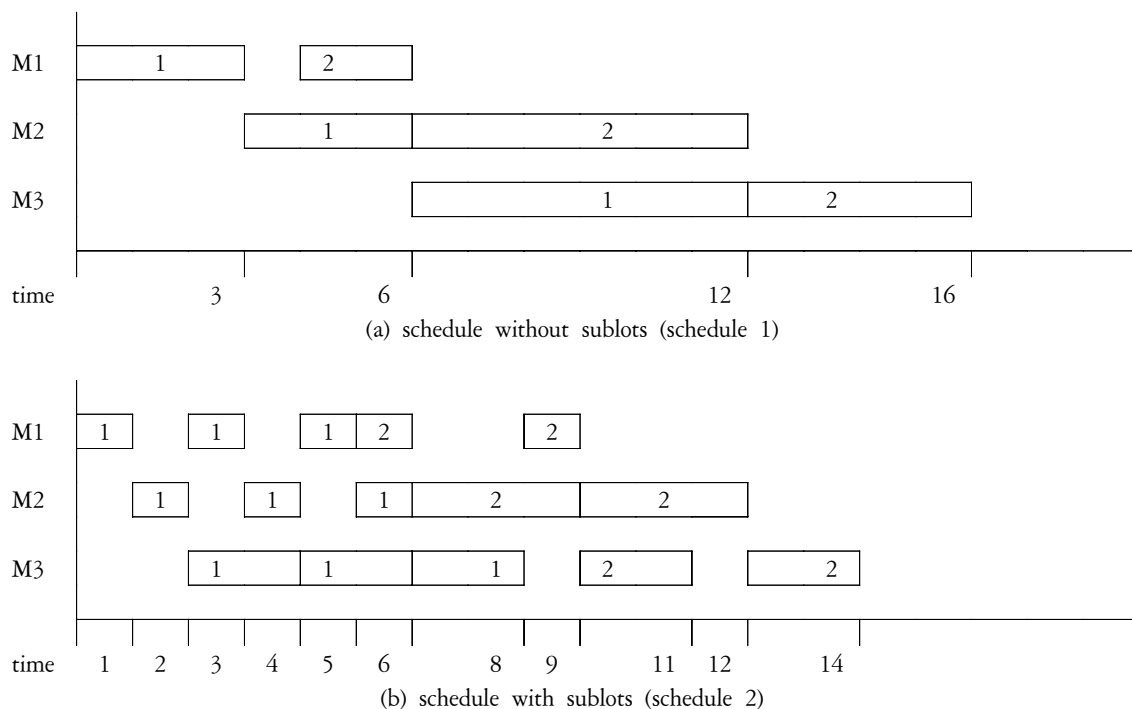


Figure 1. Two schedules for a three-machine flow shop(the numerical represent jobs).

replaces the GAs' selection and mating operators by new operators (marriage and pregnancy operators). NGA also adopts the idea of inter-chromosomal dominance to prevent the inheritance of undesirable features from sequences. With fitness values of individuals, NGA searches for the best sequences. In Section 4, some computational results are provided and the performance of NGA is compared with those of GAs. Finally, a summary of main results and conclusions are provided in Section 5.

## 2. No-wait Lot-streaming Flow Shop

The problem studied in this chapter is the minimization of the mean weighted absolute deviation of job completion times from due dates for the  $n$ -job,  $m$ -machine, no-wait lot-streaming flow shop where sublots in a job have equal-size. For job  $j$ ,  $j = 1, \dots, n$ , let  $s_j$  be the number of sublots,  $d_j$  is the due date,  $\alpha_j$  the earliness penalty,  $\beta_j$  the tardiness penalty, and  $r_{i,j}$  the subplot processing time on machine  $i$ ,  $i = 1, \dots, m$ . Let  $c_{i,j,k}$  the completion time of the subplot  $k$  of job  $j$  on machine  $i$ . Then,  $e_j = \max\{0, d_j - c_{m,j,s_j}\}$  and  $t_j = \max\{0, c_{m,j,s_j} - d_j\}$ . This problem can be formulated as a zero-one mixed integer linear programming (MILP). Let  $M$  be a big number. Define binary integer variables  $x_{j_1,j_2}$  as follows:

$$x_{j_1,j_2} = \begin{cases} 1, & \text{if job } j_1 \text{ precedes job } j_2 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$\text{minimize } z = \sum_{j=1}^n [\alpha_j e_{1j} + \beta_j t_{1j}] \quad (2)$$

$$\text{s.t. } c_{i,j_1,s_{j_1}} - r_{i,j_2} \geq c_{i,j_2,1} + M(1 - x_{j_1,j_2}), \\ \text{for } i = 1, \dots, m, j_1, j_2 = 1, \dots, n, j_1 \neq j_2 \quad (3)$$

$$c_{i,j,k} - r_{i,j} \geq c_{i,j,k-1}, \\ \text{for } i = 1, \dots, m, j = 1, \dots, n, k = 2, \dots, s_j \quad (4)$$

$$c_{i,j,k} - r_{i,j} = c_{i-1,j,k}, \\ \text{for } i = 2, \dots, m, j = 1, \dots, n, k = 1, \dots, s_j \quad (5)$$

$$c_{m,j,s_j} - t_j + e_j = d_j, \\ \text{for } j = 1, \dots, n \quad (6)$$

$$x_{j_1,j_2} + x_{j_2,j_1} = 1, \\ \text{for } j_1, j_2 = 1, \dots, n, j_1 \neq j_2 \quad (7)$$

$$c_{1,j,1} \geq r_{1,j}, \quad \text{for } j = 1, \dots, n \quad (8)$$

$$c_{i,j,k} \geq 0, \\ \text{for } i = 1, \dots, m, j = 1, \dots, n, k = 1, \dots, s_j \quad (9)$$

$$e_j, t_j \geq 0, \quad \text{for } j = 1, \dots, n \quad (10)$$

$$x_{j_1,j_2} = 0 \text{ or } 1, \\ \text{for } j_1, j_2 = 1, \dots, n, j_1 \neq j_2. \quad (11)$$

Constraint set (3) establishes the relationships between completion times of any two jobs on each machine. That is, only one job at most can be processed on each machine at the same time. Constraint set (4) states that each machine can process at most one subplot at the same time. Constraint set (5) insures that once a subplot is released from a machine, its processing on the downstream machine begins immediately. Constraint set (6) states that the completion time of a job at the last machine is equal to its due date minus or plus the amount of time it is early or tardy. Constraint set (7) insures the definition of 0-1 integer variables. That is, a job cannot precede and follow another job at the same time. Constraint set (8) states that jobs are available at time zero. The inequalities (9) and (10) insure nonnegativity of variables. The equalities (11) insure 0-1 integer values of variables.

The optimal schedule for this zero-one MILP can be determined by general purpose methods such as branch and bound enumeration or dynamic programming. Since this problem is NP-complete in the strong sense, it requires significant computational effort to solve this zero-one MILP with large  $n$ . Consequently, it is of great interest to find good approximation algorithms for the problem. The problem can also be modeled in the following standard form:

$$\text{minimize } z(\sigma_l) = \sum_{j=1}^n [\alpha_j e_{1j} + \beta_j t_{1j}], \quad (12)$$

$$\text{subject to } \sigma_l \in \Pi, \quad (13)$$

where  $\sigma_l = \{\sigma_l(1), \sigma_l(2), \dots, \sigma_l(n)\}$  is a sequence of jobs defined by a permutation of integers  $\{1, \dots, n\}$ ,  $\sigma_l(j)$  represents the  $j$ th job (also called gene) in the

sequence, and  $\Pi$  is the set of all feasible sequences. The cardinality of  $\Pi$  is  $n!$ .

Let  $c_{i,\sigma_l(j),k}$  represent the completion time of subplot  $k$  of the  $j$ th job on machine  $i$  for individual  $\sigma_l$  in the population. The optimal subplot starting and completion times for a given sequence  $\sigma_l$ ,  $l = 1, \dots, w$ , can be obtained by solving the following LP:

$$\begin{aligned} & \text{minimize } z(\sigma_l) = \sum_{j=1}^n [\alpha_j e_{1j} + \beta_j t_{1j}] \\ \text{s.t. } & c_{i,\sigma_l(j),1} - r_{i,\sigma_l(j)} \geq c_{i,\sigma_l(j-1),s_{\sigma_l(j-1)}}, \\ & \text{for } i = 1, \dots, m, j = 2, \dots, n \\ & c_{i,j,k} - r_{i,j} \geq c_{i,j,k-1}, \\ & \text{for } i = 1, \dots, m, j = 1, \dots, n, k = 2, \dots, s_j \\ & c_{i,j,k} - r_{i,j} = c_{i-1,j,k}, \\ & \text{for } i = 2, \dots, m, j = 1, \dots, n, k = 1, \dots, s_j \\ & c_{m,j,s_j} - t_j + e_j = d_j, \quad \text{for } j = 1, \dots, n \\ & c_{1,\sigma_l(1),1} \geq r_{1,\sigma_l(1)} \\ & c_{i,j,k} \geq 0, \\ & \text{for } i = 1, \dots, m, j = 1, \dots, n, k = 1, \dots, s_j \\ & e_{1j}, t_{1j} \geq 0, \quad \text{for } j = 1, \dots, n. \end{aligned}$$

### 3. New Genetic Algorithm (NGA)

With the ‘‘survival of the fittest’’ philosophy, GAs select individuals in a population to form a gene pool according to their fitness values. High fit individuals can be selected more than once, but some low fit individuals may not. Two individuals (couple) in the gene pool are randomly mated to produce two offspring for crossover and mutation processes. Unlike GAs, NGA does not have a selection process. Every individual is mated with another in a population and this mating process is called marriage. The sum of the fitness values of two mated individuals becomes the couple’s fitness value. A couple may be selected to produce a single offspring according to the couple’s fitness value and this

process is called pregnancy. Some couples may be selected multiple times and hence produce two or more offspring, while others may not. If a couple produces one offspring, its fitness value decreases by a fraction of its value (i.e., the pregnancy rate decreases), which can be interpreted as aging effect. This procedure continues until the number of offspring reaches a population size (i.e., the number of individuals in a population). The detail procedure of NGA is explained below.

NGA uses a permutation representation for individuals where a sequence of  $n$  jobs is defined by a permutation of integers  $\{1, \dots, n\}$ . NGA begins with randomly generated populations since the use of randomly generated individuals enables unbiased sampling of the search space. The optimal subplot starting and completion times for individuals are obtained by solving LP. The objective function values of the LP are transformed to obtain fitness values of individuals by the ranking procedure (Reeves, 1995). In the ranking procedure, the population is sorted according to the objective function value. An individual with the maximal objective function value has a rank of one, and an individual with the second high objective function value has a rank of 2, and so on. Ties are broken arbitrarily. These individuals’ ranks are used as their fitness values. Once every individual in a population has been mated with another, a couple is selected to produce an offspring according to the couple’s fitness value. NGA adopts the roulette wheel selection procedure (Goldberg, 1989) with couples’ fitness values. In the roulette wheel selection procedure, the ratio of a couple’s fitness value to the sum of all couples’ fitness values is the probability that this couple may be selected.

NGA adopts the idea of inter-chromosomal dominance and incorporates this idea into PMX. In NGA, PMX with inter-chromosomal dominance is developed to produce only one offspring at a time. Two offspring are generated by PMX and the offspring that has more genes from the higher fit individual in the original couple (inter-chromosomal dominance) is selected. In this way, more genes from the higher fit individual are inherited by the offspring. Suppose that  $A$  and  $B$  are the two individuals in the couple chosen for crossover, such that  $A = (5 \ 1 \ 8 \ 6 \ 7 \ 2 \ 3 \ 4)$  with a fitness value of 76 and  $B = (6 \ 8 \ 4 \ 7 \ 1 \ 3 \ 5 \ 2)$  with a fitness value of 53. Let two crossover points be 2 and 5. PMX produces two offsprings,  $A' = (5 \ 6 \ 4 \ 7 \ 1 \ 2 \ 3 \ 8)$  and  $B' = (1 \ 4 \ 8 \ 6$

7 3 5 2). PMX with inter-chromosomal dominance produces an offspring  $A'' = (5\ 6\ 4\ 7\ 1\ 2\ 3\ 8)$  since  $A''$  has more genes from higher fit individual  $A$ . Once an offspring is produced by PMX with inter-chromosomal dominance, the offspring may be mutated with certain mutation rate. NGA adopts the adjacent swap method with a constant mutation rate in which a job is exchanged with the next job in the job sequence. If the last job is to be mutated, it is exchanged with the first job in the job sequence.

### New Genetic Algorithm (NGA)

#### Step 1 (Initialization)

Generate an initial population with  $w$  individuals using a random number generator.

#### Step 2 (Calculation of Individual's Fitness)

- (a) Obtain objective function values of individuals in the population by using LP.
- (b) Compute the fitness values of individuals in the population.

#### Step 3 (Marriage and Pregnancy)

- (a) Mate every individual with another individual randomly and sum the fitness values of the couple.
- (b) Use the roulette wheel selection to choose a couple for crossover and mutation processes.

#### Step 4 (Reproduction)

- (a) Apply PMX with inter-chromosomal dominance with a constant crossover rate to the couple chosen at Step 3.
- (b) Apply the adjacent swap method with a constant mutation rate to the offspring produced by

PMX with inter-chromosomal dominance.

- (c) If the number of offspring reaches  $w$ , go to Step 5. Otherwise, go to Step 3.

#### Step 5 (Termination test)

If NGA reaches the maximum number of generations, stop. Otherwise, go to Step 2.

## 4. Computational Study

The LP formulation, NGA, and GA were coded in Visual FORTRAN with the IMSL library and ran on a Pentium IV 1.8 GHz PC. Due dates, processing times, earliness and tardiness penalties of jobs, and the number of sublots per each job, were generated according to the integer uniform distributions provided in <Table 1>.

The experiments were divided into two parts: a preliminary test and a main test. Since the performances of GA and NGA are influenced by several control parameters, a preliminary test is necessary to achieve the best parameter set for GA and NGA. In the preliminary test, 8 test problems of different sizes generated according to the data in <Table 1> were solved. GA produced the best objective function value by using a population size (PPSZ) of 100, a total of 100 generations (XGEN), a mutation rate ( $p_m$ ) of 0.01, and the stochastic remainder selection procedure without replacement (Goldberg, 1989). The best average objective function value for NGA was obtained by using a loss of pregnancy rate of  $1/PPSZ$ , a population size of 100, a total of 100 generations, and a mutation rate of 0.01. These parameters are used in the main test.

Table 1. Data used to generate test problems (all data are integers)

Data	Value
Number of jobs (NJ)	10, 15
Number of machines (NM)	2, 3, 4, 5
Number of sublots (NS)	Uniform(1, 6)
Earliness penalties	Uniform(1, 6)
Tardiness penalties	Uniform(1, 6)
Sublot processing times (SP)	Uniform(1, 31)
Job processing times	NS * SP
Due dates	Uniform(15 * NJ, 15 * (NJ + NM))

The test problems for the main test were generated in a similar way. Ten different test problems were generated for each problem size. These 80 problems were solved by NGA. NGA was applied to medium size (10 jobs and 2-5 machines) and large size (15 jobs and 2-5 machines) problems. To evaluate the performance of NGA, the solutions obtained by NGA were compared with the solutions provided by GA. The results of NGA and GA for medium and large size problems are shown in <Table 2>. The average objective function values reported in <Table 2> are the average values of three instances for each problem size. Based on these results, NGA provides an 8.15% improvement with respect to GA on the average.

## 5. Conclusions

This paper addresses the problem of minimizing the mean weighted absolute deviation of job completion times from due dates when jobs are scheduled in the  $n$ -job,  $m$ -machine, no-wait lot-streaming flow shop. This problem is formulated as a zero-one MILP. Since this problem is NP-complete in the strong sense, it requires significant computational effort to solve this zero-one MILP with large  $n$ . Consequently, it is necessary to find good approximation algorithms for the problem. NGA has been proposed to solve

the problem. The marriage and pregnancy operators have been developed and the GAs' selection and mating operators are replaced by these operators to overcome premature convergence. Also, the inter-chromosomal dominance has been introduced and incorporated into PMX to restrain the inheritance of undesired characteristics from the low fit parent. The results of NGA have been compared with that of GA. Based on these results, the solution quality provided by NGA has been improved with respect to GA for this type of problem.

## References

- Benjaafar, S. (1996), On production batches, transfer batches, and lead times, *IIE Transactions*, **28**, 357-362.
- Gershwin, S.B. (1994), *Manufacturing Systems Engineering*, Prentice Hall, New Jersey.
- Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, New York, NY.
- Graves, S.C. and Kostreva, M.M. (1986), Overlapping operations in material requirements planning, *Journal of Operations Management*, **6**, 283-294.
- Jacobs, F.R. and Bragg, D.J. (1988), Repetitive lots: Flow-time reductions through sequencing and dynamic batch sizing, *Decision Sciences*, **19**, 281-294.
- Pinedo, M. (2002), *Scheduling: Theory, Algorithms, and Systems (2nd Ed.)*, Prentice Hall, Upper Saddle River, New Jersey.
- Potts, C.N. and Van Wassenhove, L.N. (1992), Integrating scheduling with batching and lot-sizing: a review of

**Table 2.** Results for medium and large size, equal-size subplot, no-wait flow shop problems (CPU time in seconds)

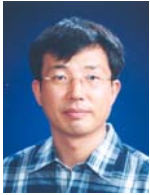
No. of Jobs	No. of machines	GA method		NGA method		%Dev ( $(z_g - z_n/z_g) \times 100$ )
		Avg. obj. value ( $z_g$ )	CPU time	Avg. obj. value ( $z_n$ )	CPU time	
10	2	5046.30	2002.63	4801.80	2615.47	4.85
	3	5353.80	2327.81	5169.70	2959.70	3.44
	4	6348.70	2694.46	6158.50	3286.54	3.00
	5	6524.90	3027.40	6268.60	3625.31	3.93
15	2	7521.10	2208.89	6405.50	2819.59	14.83
	3	10007.00	2653.51	8728.90	3257.78	12.77
	4	11423.70	3225.48	10225.60	3854.39	10.49
	5	12802.60	3745.68	11275.90	4364.83	11.92
Average		8128.51		7379.31		8.15

algorithms and complexity, *Journal of the Operational Research Society*, 43, 395-406.

Reeves, C.R. (1993), *Modern Heuristic Techniques for Combinatorial Problems (Ed.)*, Halsted Press, New York.

Reeves, C.R. (1995), A genetic algorithm for flowshop sequencing, *Computers and Operations Research*, 22(1), 5-13.

Sriskandarajah, C. and Wagneur, E. (1999), Lot streaming and scheduling multiple products in two-machine no-wait flowshops, *IIE Transactions*, 31, 695-707.



### 윤석훈

서울대학교 산업공학 학사

미국 North Carolina A&T State University 산업  
공학 석사

Pennsylvania State University 산업공학 박사

현재: 숭실대학교 산업정보시스템공학과 교수

관심분야: Scheduling, Mathematical Programming,  
Applied Optimization, Network Design and  
Optimization 등