Journal of the Korean Institute of Industrial Engineers
Vol. 30, No. 2, pp. 159-164, June 2004.

Lot-Streaming Flow Shop Problem with Delivery Windows

Suk-Hun Yoon†

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

딜리버리 윈도우 로트-스트리밍 흐름 공정 문제

윤 석 훈

숭실대학교 산업정보시스템공학과

Lot-streaming is the process of splitting a job (lot) into a number of smaller sublots and then scheduling these sublots in order to accelerate the completion of jobs in a multi-stage production system. A new genetic algorithm (NGA) is proposed for a n-job, m-machine, equal-size sublot lot-streaming flow shop scheduling problem with delivery windows in which the objective is to minimize the mean weighted absolute deviation of job completion times from due dates. The performance of NGA is compared with that of an adjacent pairwise interchange (API) method and the results of computational experiments show that NGA works well for this type of problem.

***Keywords*:** scheduling, flow shop, lot-streaming, mean weighted absolute deviation, genetic algorithms

1. Introduction

The concept of delivery windows (due date tolerances) has stemmed from the fact that some customers impose service deadlines and earliest service time constraints(Ventura and Weng, 1996). Delivery windows arises in a variety of applications, including retail distribution, mail and newspaper delivery, municipal waste collection, fuel oil delivery, school bus routing, airline and railroad scheduling, trucking and bargeline fleet scheduling, and demand responsive bus systems (Solomon and Desrosiers, 1988). In machine scheduling problems concerning delivery windows, a job will incur no penalty if it completes within its delivery window (Weng and Ventura, 1995). Earliness (tardiness) is measured only from the left (right) side of the delivery window. Figure 1 shows a continuous penalty function, where for job j, $j = 1, ..., n$, d_j is the due date, α_j the earliness penalty, β_j the tardiness penalty, v_j left (right) delivery window side, e_{1j} earliness with penalties, t_{1j} tardiness with penalties, e_{2j} earliness without penalties, and t_{2j} tardiness without penalties.

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 (Baker and Pike, 1990). The idea of lot-streaming scheduling has been introduced by Reiter (1966) and is consistent with the optimal production technology (OPT) concept (Glass *et al*., 1996). The use of sublots usually results in substantially shorter job completion times for the corresponding schedule. This process is illustrated by the two-job, three-machine, equal-size sublot lot-streaming flow shop with infinite capacity buffers shown in <Figure 2>, 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 (M1), 3 and 6 time units on machine 2 (M2), and 6 and 4 time units on machine 3 (M3), respectively. The due dates of jobs 1 and 2 are in 9 and 11 time units. A job that completes within 1 time unit from its due date does not incur penalty. If the jobs are not split into sublots, the

This work was supported by the Soongsil University Research Fund.
† Corresponding author : Professor 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 April 2004; revision received May 2004; accepted May 2004.

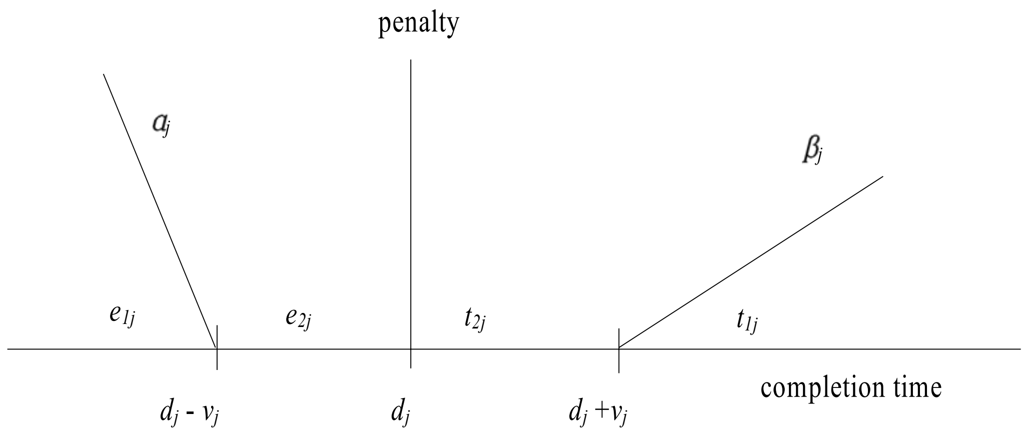
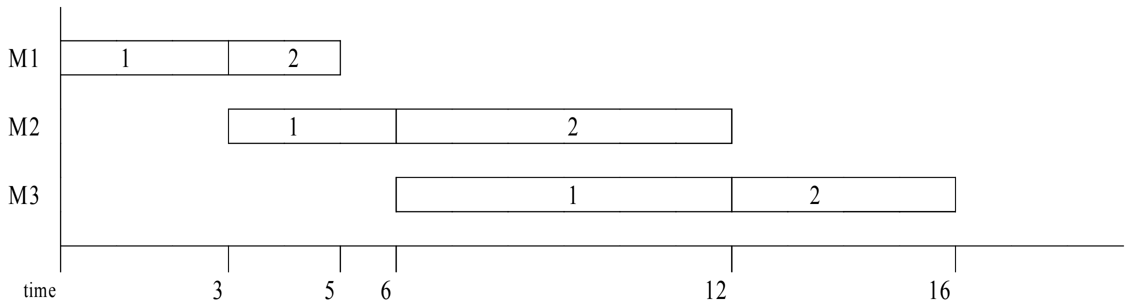
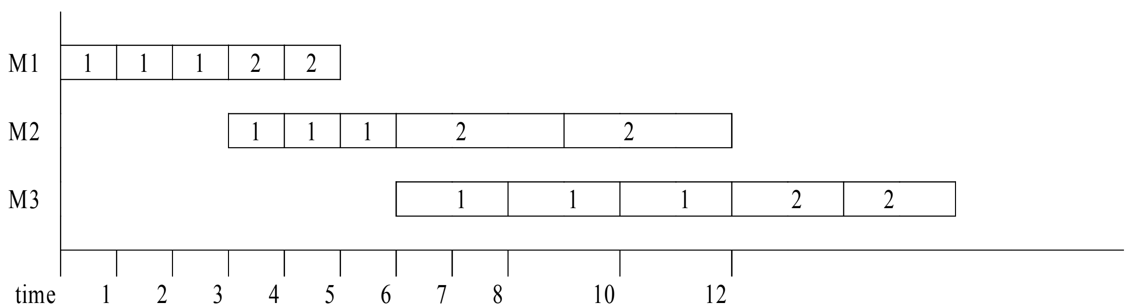


Figure 1. A continuous penalty function for due date tolerances.



(a) schedule without sublots (schedule 1)



(b) schedule with sublots (schedule 2)

Figure 2. Two schedules for a three-machine flow shop (the numerical represent jobs).

completion times of jobs 1 and 2 will be 12 and 16 time units, and both jobs will have a delay of 2 and 4 time units from the delivery window, respectively (schedule 1). As <Figure 2(b)> shows, when the jobs are split into sublots, the completion times of jobs are reduced to 8 and 12 time units, respectively, and both jobs can complete within its delivery window (schedule 2).

The problem addressed in this paper is the minimization of the mean weighted absolute deviation of job completion times from due dates with tolerances for the n-job, m-machine, equal-size sublot, lot-streaming flow shop problem with delivery windows. Since this problem is NP-complete (Hall and Posner, 1991; Garey and Johnson, 1979), it requires significant computational effort to obtain solutions with large n. Consequently, it is of great interest to find good approximation algorithms for the problem. For a given job sequence, the insertion of idle times between sublots and between jobs may improve the objective function in some cases. In Section 2, a linear programming (LP) formulation is presented for the equal-size sublot flow shop problem with delivery windows to find the optimal completion times of sublots for a given job sequence (individual). The objective function values of the LP are transformed to obtain fitness values of individuals. With these fitness values, NGA searches for the best sequences. In Section 3, the functions of the NGA are defined and explained in detail. In Section 4, the procedure to generate sample problems and the results of the computational experiments are provided. The performance of the NGA approach is compared with that of the API method. Finally, summary of main results and conclusions are provided in Section 5.

2. Lot-Streaming Flow Shop Problem with Delivery Windows

For job j, j = 1, ..., n, let s_j be the number of sublots, $p_{I,j}$ the processing time on machine i, and $r_{i,j}$ the sublot processing time on machine i. If the completion time of the sublot k of job j on machine i is $c_{i,j,k}$, then e_{1j} = max{0, d_j - v_j - $c_{m,j,sj}$}, t_{1j} = max{0, $c_{m,j,sj}$ - (d_j + v_j)}, e_{2j} = min[max{0, d_j - $c_{m,j,si}$}, v_j], and t_{2i} = min[max{0, $c_{m,j,sj}$ - d_j}, v_j]. Let σ = {σ(1), σ(2), ..., σ(n)} represent a sequence of jobs defined by a permutation of integers {1,⋯, n}, where σ(j) represents the jth job in the sequence and the number of all feasible sequences is n!. Let $c_{i,\sigma(j),k}$ represent the completion time of sublot k of the jth job on machine i in a given job sequence σ. For a given job sequence σ, the problem can be formulated as follows:

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

$$s.t. \quad c_{i,\sigma(j),1} - r_{i,\sigma(j)} \geq c_{i,\sigma(j-1),s_{\sigma(j-1)}}, \quad \text{for } i=1,...,m,\ j=2,...,n \tag{2}$$

$$c_{i,j,k} - r_{i,j} \geq c_{i,j,k-1}, \quad \text{for } i=1,...,m,\ j=1,...,n,\ k=2,...,s_j \tag{3}$$

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

$$c_{m,j,s_j} - t_{1j} - t_{2j} + e_{1j} + e_{2j} = d_j, \quad \text{for } j=1,...,n \tag{5}$$

$$c_{1,\sigma(1),1} \geq r_{1,\sigma(1)} \tag{6}$$

$$0 \leq e_{2j} \leq v_j, \quad \text{for } j=1,...,n \tag{7}$$

$$0 \leq t_{2j} \leq v_j, \quad \text{for } j=1,...,n \tag{8}$$

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

$$e_{1j}, t_{1j} \geq 0, \quad \text{for } j=1,...,n. \tag{10}$$

Constraint set (2) 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 (3) states that each machine can process at most one sublot at the same time. Constraint set (4) insures that each sublot on the current machine cannot transfer to the next machine before its processing is finished. Constraint set (5) 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 (6) states that jobs are available at time zero. The inequalities (7) and (8) insure the lower and upper bound of the earliness and tardiness variables inside due date tolerances. The inequalities (9) and (10) insure nonnegativity of variables.

3. New Genetic Algorithm (NGA)

Genetic algorithms(GAs) have been successfully applied to various problems that could not have been readily solved with conventional computational techniques(Liepins and Hilliard, 1989). Although there are many possible variants of the basic GAs, the fundamental underlying mechanism operates on a population of individuals and consists of three operations (selection, crossover and mutation) to construct new solutions from individuals of the current population. Individuals are chosen on the basis of their fitness values according to the GA philosophy of survival-of-the-fitness. Based on this selection strategy, some individuals may be selected more than once, while other individuals may never be selected. Thus, the performance of GAs is highly sensitive to the selection process. GAs sometimes end in premature convergence using the proportionate selection scheme (Srinivas and Patnaik, 1994).

NGA replaces the selection and mating operators of GAs by new operators (marriage and pregnancy operators). Every individual is mated only once with another in a population and this mating process is called marriage. An individual's fitness value is calculated using fitness function by rank (Reeves, 1995). The sum of two mated individuals' fitness values becomes the couple's fitness value. A couple may be selected to produce an offspring according to the couple's fitness value (called pregnancy rate). Some couples may produce two or more offspring, while others may not. If a couple has produced one offspring, its fitness value decreases by some proportion of its fitness value (i.e., the pregnancy rate decreases), which can be interpreted as aging effect.

NGA adopts the idea of inter-chromosomal dominance and incorporate this idea into partially matched crossover (PMX). Yoon and Ventura (2002) explains the procedure of PMX in detail. Since a couple produces only one offspring by this PMX with inter-chromosomal dominance, the choice of an individual in a couple is important. PMX with inter-chromosomal dominance has two stages: (1) crossover between two individuals using PMX to generate two offspring, (2) selection of an offspring, which has more genes from the higher fit individual than the other (inter-chromosomal dominance). In this way, more genes from the high fit individual will be inherited to the offspring. The procedure of PMX with inter-chromosomal dominance is illustrated by two individuals A and B in the couple chosen for crossover such that A = (10 6 5 1 9 7 8 2 3 4) with fitness value of 76 and B = (3 7 6 10 5 8 1 4 9 2) with fitness value of 53 in <Figure 3>. Suppose that two crossover points are 3 and 7. PMX produces two offspring A' = (7 6 9 10 5 8 1 2 3 4) and B' = (3 10 6 1 9 7 8 4 5 2) as shown in <Figure 3(b)>. PMX with inter-chromosomal dominance produces an offspring A' = (7 6 9 10 5 8 1 2 3 4) since A' has more genes from high fit individual A as shown in <Figure 3(c)>.

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 values of individuals in the population by using LP.
(b) Compute the fitness values of individuals in the population using the formula

$$p([k]) = \frac{2k}{w(w+1)}, \qquad \text{for } k = 1, \ldots, w,$$

where $[k]$ is the kth individual in descending order of the objective function value.

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(Goldberg, 1989) to choose a couple for crossover and mutation processes.

Step 4 (Reproduction)
(a) Apply PMXt crossover rate to the couple chosen at Step 3.
(b) Apply the adjacent swap method with a constant mutation rate to the offspring 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.

				crossover point 1				crossover point 2		
individual A:	10	6	5	1	9	7	8	2	3	4
individual B:	3	7	6	10	5	8	1	4	9	2

(a) Two individuals before crossover

individual A':	7	6	9	10	5	8	1	2	3	4
individual B':	3	10	6	1	9	7	8	4	5	2

(b) Two individuals after PMX operation

individual A':	7	6	9	10	5	8	1	2	3	4

(c) One individual after PMX with inter-chromosomal dominance operation

Figure 3. Comparison between PMX and 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 API method were coded in Visual FORTRAN with the IMSL library and ran on a Pentium IV 1.8 GHz PC. Since no sample problems were found in the literature, the test problems were generated randomly for the delivery window length 6, 12, and 24 (Hall and Posner, 2001). The size of each problem is represented by the number of jobs (*NJ*) and the number of machines (*NM*). The number of sublots and the earliness and tardiness penalties of jobs are uniformly distributed in [1, 6]. Sublot processing times and due dates of jobs are uniformly distributed in [1, 31] and [15 * *NJ*, 15 * (*NJ* + *NM*))], respectively.

The experiments were divided into two parts: preliminary test and main test. A preliminary test was performed to achieve the best control parameter sets that influence the performance of NGA. In preliminary test, 12 test problems of different sizes generated according to the above data were solved and the best average objective value was obtained by using a population size (PPSZ) of 100, a total of 100 generations, the fitness function by rank, a loss of pregnancy rate of 1/PPSZ, and a mutation rate of 0.01.

Table 1. Results for medium and large size, equal-size sublot, delivery window problems

(a) Window length of six

Window length	No. of Jobs	No. of machines	API method Avg. objective (z_h)	NGA method Avg. objective (z_n)	%Dev (z_h-z_n/z_h)×100
6	10	2	5891.67	3987.00	32.33
		3	4763.00	3717.33	21.95
		4	4678.33	4136.33	11.59
		5	4386.33	4024.33	8.25
	15	2	6833.33	6030.33	11.75
		3	7882.33	5755.33	26.98
		4	9460.33	6925.67	26.79
		5	9510.00	6375.33	32.96
Average			6675.67	5118.96	23.32

(b) Window length of twelve

Window length	No. of Jobs	No. of machines	API method Avg. objective (z_h)	NGA method Avg. objective (z_n)	%Dev (z_h-z_n/z_h)×100
12	10	2	4128.00	3910.00	5.28
		3	4691.67	3646.00	22.29
		4	4475.67	4062.00	9.24
		5	4311.33	3954.33	8.28
	15	2	6713.00	5873.33	12.51
		3	7748.33	5555.33	28.30
		4	9345.00	6796.00	27.28
		5	9380.00	6274.33	33.11
Average			6349.13	5008.92	21.11

These control parameters are used in the main test.

The test problems for the main test were generated in a similar way. Three different test problems were generated for each problem size. These 36 problems were solved by NGA for the delivery window length 6, 12, and 24. For small size lot-streaming flow shop problems (7 jobs and 2-5 machines), the results of the NGA were compared with the optimal solutions obtained by exhaustive search. NGA achieved optimal solutions for all twelve small size problems.

NGA was also 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 average objective function value of NGA is compared with that of API method, which is known to be a good heuristic for the single machine scheduling problem with the mean tardiness performance measure (Baker, 1974; Wagner and Ragatz, 1994). The results of NGA and API for medium and large size problems are shown in <Table 1>. The average objective function values reported in <Table 1> are the average values of three instances for each problem size. Based on these results, NGA provides 23.32%, 21.11%, and 21.36% better solutions than API method on the average for delivery window length 6, 12, and 24, respectively.

5. Conclusions

This paper has addressed the *n*-job, *m*-machine, equal-size sublot, lot-streaming flow shop problem with delivery windows, in which the objective is to minimize the mean weighted absolute deviation from the due dates with tolerances. Meta-heuristic algorithms such as GAs have been used for many scheduling problems. However, GAs have sometimes experienced premature convergence(Beasley *et al.*, 1993). NGA has been proposed to overcome the premature convergence by substituting marriage and pregnancy operators for GAs' selection and mating operators. The performance of NGA was compared with that of the API method and the results of the computational experiments show that NGA works well for this type of problem.

References

Baker, K.R. and Pyke, D.F. (1990), Solution procedures for the lot-streaming problem, *Decision Sciences*, **21**, 475-491.

Baker, K.R. (1974), *Introduction to Sequencing and Scheduling*, Wiley, New York.

Beasley, D., Bull, D.R., and Martin, R.R. (1993),An overview of genetic algorithms: part 1, fundamentals, *University Computing*, **15**, 58-69.

Garey, M.R. and John, D.S. (1979), *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, CA.

Glass, C.A., Gupta, J.N.D., and Potts, C.N. (1994), Lot streaming in three-stage production processes, *European Journal of Operational Research*, **75**, 378-394.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, New York, NY.

Hall, N.G. and Posner, M.E. (1991), Earliness-tardiness scheduling problems, I: weighted deviation of completion times about a common due date, *Operations Research*, **39**(5), 836-846.

Hall, N.G. and Posner, M.E. (2001), Generating experimental data for computational testing with machine scheduling applications, *Operations Research*, **49**, 854-865.

Liepins, G.E. and Hilliard, M.R. (1989), Genetic algorithms: foundation and applications, *Annals of Operations Research*, **21**, 31-58.

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

Reiter, S. (1966), A system for managing job-shop production, *The Journal of Business*, **34**, 371-393.

Solomon, M.M. and Desrosiers, J. (1988), Time window constrained routing and scheduling problems, *Transportation science*, **22**(1), 1-13.

Srinivas, M. and Patnaik, L.M. (1994), Genetic algorithms: a survey, *Computer*, **27**, 17-26.

Ventura, J.A. and Weng, M.X. (1996), Single machine scheduling with a common delivery window, *Journal of the Operational Research Society*, **47**, 424-434.

Wagner, B.J. and Ragatz, G.L. (1994), The impact of lot splitting on due date performance, *Journal of Operations Management*, **12**, 13-25.

Weng, M.X. and Ventura, J.A. (1995), Single-machine earliness-tardiness scheduling about a common due date with tolerances, *International Journal of Production Economics*, **42**, 217-227.

Yoon, S.-H. and Ventura, J.A. (2002), An application of genetic algorithms to lot-streaming flow shop scheduling, *IIE Transactions*, **34**(9), 779-787.