

On Lot-Streaming Flow Shops with Stretch Criterion

Suk-Hun Yoon[†]

Department of Industrial and Information Systems Engineering, Soongsil University

로트 스트리밍 흐름공정 일정계획의 스트레치 최소화

윤 석 훈[†]

승실대학교 산업정보시스템공학과

Lot-streaming is the process of splitting a job (lot) into sublots to allow the overlapping of operations between successive machines in a multi-stage production system. A new genetic algorithm (NGA) is proposed for an n -job, m -machine, lot-streaming flow shop scheduling problem with equal-size sublots in which the objective is to minimize the total stretch. The stretch of a job is the ratio of the amount of time the job spent before its completion to its processing time. NGA replaces the selection and mating operators of genetic algorithms (GAs) by marriage and pregnancy operators and incorporates the idea of inter-chromosomal dominance and individuals' similarities. Extensive computational experiments for medium to large-scale lot-streaming flow-shop scheduling problems have been conducted to compare the performance of NGA with that of GA.

Keywords : Scheduling, Flow Shop, Lot-Streaming, Total Stretch, Genetic Algorithms

1. Introduction

In many scheduling problems, the focus of performance measure has been on the flow time, which is defined as the amount of time that a given job spends in the system. Recently, alternative performance measures have been considered and among them, the stretch measure has received a lot of attention [3]. The stretch of a job formally is defined as the ratio of its flow time to its required processing time [1]. The stretch measure relates the jobs' waiting times to their processing times, and may reflect users' psychological expectation that, in a system with highly various job sizes, users are willing to wait longer for larger jobs [14].

This paper presents a solution methodology for the n -job,

m -machine, lot-streaming flow shop scheduling problem with equal-size sublots in which the objective is to minimize the total stretch. Since the problem of minimizing the total stretch with different job release times is not tractable even for a single machine [16], it is usually very difficult to find optimal or near optimal schedules by general local search methods such as the adjacent pairwise interchange method. Recently, meta-heuristic methods, such as genetic algorithms, simulated annealing, and tabu search, have been successful in solving combinatorial optimization problems [7].

GAs are stochastic search methods designed to search large and complex spaces by exploitation of currently known solutions and a robust exploration of the space. GAs start with a collection (called population) of randomly selected solutions (called individuals, strings, or chromosomes). In each iteration (called generation), the fitness of every individual is evaluated. With the "survival of the fittest" philosophy, GAs select individuals from a population to form a

gene pool according to their fitness values. Some individuals may be selected more than once, while other individuals may never be selected. The individuals in the gene pool are mated randomly and the mated individuals (couples) go through crossover and mutation operators to form offspring for a new population. For scheduling problems, a partially matched crossover operator (PMX) and an adjacent swap method have frequently been used for the crossover and mutation operators, respectively [8]. The algorithm terminates after a pre-specified number of generations have been produced or a satisfactory solution has been reached.

In spite of the continued growth and refinement, GAs can still fail for a variety of reasons, including failure to represent problem specific information, and convergence to local optima (premature convergence). The problem of premature convergence is related with the loss of genetic diversity of the population, which can be the cause of poor quality of the solutions [11]. In this paper, NGA is proposed to prevent the premature convergence through the new idea and operators. NGA replaces the GAs' selection and mating operators by new operators (marriage and pregnancy operators). NGA also combines the idea of inter-chromosomal dominance and individuals' similarity to prevent the inheritance of undesirable features.

Following the literature review in the next section, the problem of minimizing the total stretch in the n -job, m -machine lot-streaming flow shop scheduling is defined in section 3. The development of NGA and its application to the lot-streaming flow shop scheduling problem are presented in section 4. In section 5, the results of extensive computational experiments comparing NGA with GA are provided. Finally, a summary of the main results and conclusions are provided in section 6.

2. Literature Review

The potential benefits of lot-streaming in batch manufacturing are pointed out in [10, 12]. These benefits include reduction of production lead times, reduction of work-in-process inventory and costs, reduction of interim storage and space requirements, reduction of material handling system capacity requirements, and improvement of product delivery. An extensive review of research in this area is presented in [4, 5].

Zhang et al. [18] address the multi-job lot-streaming prob-

lem in two-stage hybrid flow shops with m identical machines at the first stage and a single machine at the second stage to minimize the mean completion time. They provide the mixed integer linear programming formulation and a lower bound. Marimuthu et al. [13] address the n -job, m -machine, lot-streaming flow shop problem to minimize the makespan. They propose genetic algorithm and hybrid genetic algorithm to obtain best sequences. Tseng and Liao [17] address the n -job, m -machine, lot-streaming flow shop problem to minimize the total weighted earliness and tardiness. They propose a new discrete particle swarm optimization algorithm incorporating the net benefit of movement algorithm to search for the best sequence. Pan et al. [15] propose a discrete artificial bee colony algorithm to solve the lot streaming flow shop scheduling problem with the criterion of total weighted earliness and tardiness penalties.

Defersha and Chen [6] address the lot streaming job shop with routing flexibility, sequence-dependent setups, machine release dates and lag time. They propose an island-model parallel genetic algorithm. Buscher and Shen [2] consider the lot streaming problem in a job shop environment, with consistent sublots. They present three-phase algorithm that incorporates the predetermination of subplot sizes, the determination of schedules based on tabu search, and the variation of subplot sizes. Hall et al. [9] address the no-wait two machine, lot-streaming open shop problem to minimize the makespan. They formulate the problem as a classical traveling salesman problem with a pseudo-polynomial number of cities and present a dynamic programming algorithm to generate all the dominant schedules for each job.

3. Minimization of the total stretch in the n -job, m -machine Flow Shop

There is a set of jobs $J = \{1, 2, \dots, n\}$ that has to be processed in the system consisting of m machines in series. Each machine processes jobs in the same order. For job j , $j = 1, \dots, n$, let s_j be the number of sublots, τ_j the release time, p_{ij} the processing time on machine i , $r_{i,j}$ ($= p_{i,j}/s_j$) the subplot processing time on machine i , and $c_{i,\sigma(q),k}$ represent the completion time of subplot k of the q th job on machine i . Let $p_j = \sum_{i=1}^m p_{i,j}$ represent the sum of processing times of job j . Then the total stretch is defined as $\sum_{j=1}^n (c_{m,j,s_j} - \tau_j) / p_j$. For a given sequence σ , the problem can be formulated as follows:

$$\text{minimize } z(\sigma) = \sum_{j=1}^n \frac{c_{m,j,s_j} - \tau_j}{p_j}; \quad (1)$$

Subject to

$$c_{i,\sigma(j),1} - r_{i,\sigma(j)} \geq c_{i,\sigma(j-1),s_{\sigma(j-1)'}} \\ \text{for } i = 1, \dots, m, \quad j = 2, \dots, n; \quad (2)$$

$$c_{i,j,k} - k_{i,j} \geq c_{i,j,k-1'} \\ \text{for } i = 1, \dots, m, \quad j = 1, \dots, n, \quad k = 2, \dots, s_j \quad (3)$$

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

$$c_{1,\sigma(j),1} \geq r_{1,\sigma(j)} + \tau_j \quad \text{for } j = 1, \dots, n; \quad (5)$$

Constraint set (2) establishes the relationships between completion times of adjacent jobs on each machine and assures that a machine can process at most one job at the same time. Constraint set (3) provides the relationships between completion times of adjacent sublots in each job. That is, each machine can process at most one subplot at the same time. Constraint set (4) insures that each subplot on the current machine cannot be transferred to the next machine before its processing is finished. Constraint set (5) states that all jobs are available at their release times.

4. New Genetic Algorithm(NGA)

Unlike GAs, NGA does not have a selection process. Every individual in the population is mated with another individual to form a couple 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 an offspring according to the couple's fitness value and this process is called pregnancy. If a couple produces an offspring, its fitness value decreases by a fraction of its value that is called a loss of pregnancy rate, which can be interpreted as the aging effect. Some couples may be selected multiple times and produce two or more offspring, while others may not. This procedure continues until the number of offspring reaches the population size. The detail procedure of NGA is explained below.

NGA generates two types of individuals in an initial population that are randomly generated individuals and seed individuals. Seed individuals are generated by several pairwise interchange methods such as the non-adjacent pairwise

interchange, the extraction and forward shifted reinsertion, and the extraction and backward shifted reinsertion. Balancing randomly generated individuals and seed individuals enhances search for unbiased sampling of the space, which provides robust exploration and exploitation of good quality of solutions.

Objective function values of individuals are transformed to their fitness values. Let z_{max} , z_{min} and z_{avg} be the maximum, minimum and average objective values in the current population, respectively, and $z(\sigma_l)$ is the objective value of individual l . In NGA, two types of transformations are considered: normalization procedure and ranking procedure. The normalized fitness value of individual l (f_{norm}) can be calculated by the following equation:

$$f_{norm}(\sigma_l) = (z_{max} - z(\sigma_l) + z_{min}) / z_{avg}, \quad \text{for } l = 1, \dots, w,$$

where w is the population size. In the ranking procedure, the population is sorted according to the objective function value. An individual with the maximal objective function value has fitness value of one, and an individual with the second highest objective function value has fitness value of 2, and so on.

Two types of individuals' marriages are considered: random mating (love marriage) and mating between similar individuals (arranged marriage). NGA defines individuals' similarity as the difference between individuals' fitness values. In the scheme of arranged marriage, two individuals are mated where their fitness values are closest in a population. After every individual in a population has been mated with another to construct a mating pool, a couple from the mating pool is selected according to the couple's fitness value. NGA adopts the roulette wheel selection procedure with couples' fitness values. The probability that a couple may be selected is the ratio of the couple's fitness value to the sum of all couples' fitness values.

The selected couple goes through the PMX procedure with inter-chromosomal dominance to produce only one offspring at a time. Firstly, two crossover points are picked randomly. The genes between these two crossover points define a match table and are exchanged. The genes before the first crossover point and after the second one are exchanged if the genes are included in the match table. Secondly, out of these two offspring, the one that has more genes from the higher fit individual in the original couple is selected (inter-chromosomal dominance). Instead of comparing genes, if the distance between the two crossover points is greater than the half

length of an individual, the offspring from the lower fit individual is selected; otherwise, the offspring from the higher fit individual is selected. In this way, more genes from the higher fit individual are inherited by the offspring. For example, suppose that A and B are the individuals in the couple chosen for crossover, such that $A = (9\ 7\ 3\ 2\ 6\ 4\ 5\ 8\ 1)$ with a fitness value of 57 and $B = (8\ 4\ 5\ 7\ 2\ 3\ 9\ 1\ 6)$ with a fitness value of 20, and the two crossover points are 2 and 7. Firstly, swap the genes between two crossover points (3, 2, 6, 4, 5 of A and 5, 7, 2, 3, 9 of B) and construct the match table comparing the genes in the same positions (loci) between two crossover points. The comparison (3 \leftrightarrow 5, 2 \leftrightarrow 7, 6 \leftrightarrow 2, 4 \leftrightarrow 3, 5 \leftrightarrow 9) results in the match table (6 \leftrightarrow 7, 4 \leftrightarrow 9). According to the table, if genes have the value 6, 7, 4, and 9, they are exchanged into 7, 6, 9, and 4, respectively. Two offspring $A' = (4\ 6\ 5\ 7\ 2\ 3\ 9\ 8\ 1)$ and $B' = (8\ 9\ 3\ 2\ 6\ 4\ 5\ 1\ 7)$ are produced. Secondly, since the distance between the two crossover points ($7-2 = 5$) is greater than the half length of an individual ($9/2 = 4.5$), the offspring from the lower fit individual is selected. Thus, PMX with inter-chromosomal dominance produces the offspring $B'' = (8\ 9\ 3\ 2\ 6\ 4\ 5\ 1\ 7)$.

Once an offspring is produced by PMX with inter-chromosomal dominance, the offspring is mutated using 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 0 (Initialization)

In a preliminary test, the best set of following parameters is determined before the main test: determination of fitness values (f_{scale} , f_{rank}), the marriage function, the loss of pregnancy rate, population size (w), number of generations (GEN), crossover probability (P_c), mutation probability (P_m), number of seed selection individuals (N_s)

Step 1 (Construction of an initial population)

- (a) Generate an initial population with N_s seed selection individuals.
- (b) Generate an initial population with $w-N_s$ individuals using a random number generator.

Step 2 (Calculation of Individual's Fitness)

- (a) Obtain objective function values of individuals in the

population.

- (b) Compute the fitness values of individuals by normalization or rank.

Step 3 (Marriage)

- (a) Mate every individual with another individual randomly or considering individual similarities
- (b) Compute the fitness values of the couple.

Step 4 (Pregnancy)

Use the roulette wheel selection to choose a couple for Birth.

Step 5 (Birth)

- (a) Apply PMX with inter-chromosomal dominance with a crossover rate to the couple chosen at Step 4.
- (b) Decrease the fitness of the couple by a loss of pregnancy rate.
- (c) Apply the adjacent swap method with a mutation rate to the offspring produced by PMX with inter-chromosomal dominance.
- (d) If the number of offspring reaches w , go to Step 6. Otherwise, go to Step 4.

Step 6 (Termination test)

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

GAs sometimes end in premature convergence using a proportionate selection scheme. The crossover procedure is performed to produce two offspring. One offspring may have high fitness value, but the other has relatively low fitness value. By introducing the new operators and idea, NGA precludes these two drawbacks that are premature convergence and inheritance of characteristics from low fit individuals. First, since NGA uses all individuals in a population and replaces the selection and mating operators by the marriage and pregnancy operators, extraordinary individuals with high fitness value cannot dominate the population in the early generation. Once a couple gives birth to an offspring, the probability that the couple is selected for the next birth decreases by reducing the couple's fitness value. Thus, the desirable characteristics of high fit individuals are inherited by the next generation. Second, the PMX with inter-chromosomal dominance produces only one offspring, in which more genes come from the high fit parent than from the other parent. Thus, the inheritance of undesired characteristics from the low fit parent is restrained.

5. Computational Study

NGA and GA were coded in Visual FORTRAN and ran on an Intel Core i7 CPU@3.4 GHz PC. The test problems were generated randomly, since no sample problems were found in the literature that could be used as a benchmark for testing the proposed NGA. Processing times, release times of jobs, and the number of sublots per each job, were generated according to the integer uniform distributions provided in <Table 1>.

<Table 1> Data Used to Generate Test Problems(All Data are Integers)

Data	Value
Number of jobs	5, 7, 10, 15, 20, 30
Number of machines	2, 3, 4, 5
Number of sublots (NT)	Uniform (1, 6)
sublot processing times (SP)	Uniform (1, 31)
Job processing times	NT×SP
Job release times	Uniform (1, 6)

The experiments were divided into two parts : a preliminary test and a main test. The preliminary test is necessary to determine the best set of parameters for each algorithm, since the performances of GA and NGA are influenced by several control parameters. In the preliminary test, 5 test problems of different sizes were generated according to all combinations of number of jobs and number of machines provided in <Table 1>. Each problem was solved by both algorithms with different combinations of control parameters. The control parameters of GA investigated in the preliminary test are the fitness function, the population size (PPSZ), the number of generations (XGEN), the crossover rate (*Pc*), and the mutation rate (*Pm*). The control parameters of NGA are the fitness function, the marriage function, the loss of pregnancy rate, the number of seed individuals generated for an initial population (*Ns*), PPSZ, XGEN, *Pc*, and *Pm* that provide the best performance. The best choices of the control parameters found in the preliminary test for each algorithm are as follows: For GA, they are the stochastic remainder selection procedure without replacement, the fitness function by rank, a population size of 100, a total of 100 generations, PMX with a crossover rate of 1.0, and the adjacent swap method with a mutation rate of 0.01. For NGA, they are the fitness function by rank, a random marriage, a loss of pregnancy rate of 1/PPSZ, a total of 10 seed individuals, a population size of 100, a total of 100 generations, a cross-

over rate of 1.0, and a mutation rate of 0.01. These parameters were used in the main test.

The test problems for the main test were generated in a similar way. Nine different test problems were generated for each problem size. These 216 problems were solved by NGA. For small size lot-streaming flow shop problems (5 and 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 seventy two small size problems.

NGA was also applied to medium size (10 and 15 jobs and 2~5 machines) and large size (20 and 30 jobs and 2~5 machines) problems. To evaluate the performance of NGA, the solutions obtained by NGA were compared with those by GA. The results of NGA and GA are shown in <Table 2>. The average objective function values reported in <Table 2> are the average values of nine instances for each problem size. Based on these results, NGA yields 6.21, 8.46, 15.32 and 15.29% higher performance for 10, 15, 20, and 30 jobs in regard to GA, respectively.

<Table 2> Results for Medium and Large Size Problems

No. of Jobs	No. of machines	GA method	NGA method	%Dev ($z_0 - z_n / z_0$)x100
		Avg. obj. value (zg)	Avg. obj. value (zn)	
10	2	8.98	8.30	7.50
	3	13.57	12.65	6.78
	4	18.80	17.63	6.21
	5	24.10	23.05	4.36
15	2	12.15	10.56	13.07
	3	19.55	17.91	8.36
	4	27.04	24.78	8.35
	5	30.91	29.66	4.06
20	2	15.52	11.85	23.66
	3	24.75	20.68	16.47
	4	31.71	28.95	8.69
	5	40.91	35.80	12.48
30	2	24.60	17.22	30.02
	3	32.87	27.43	16.54
	4	45.53	43.13	5.27
	5	58.71	53.23	9.34
Average		26.86	23.93	11.32

6. Conclusions

In this paper, NGA has been proposed to avoid the two drawbacks of GAs, which are premature convergence and

inheritance of characteristics from individuals with low fitness values. Instead of using a selection strategy based on the individual's fitness value, NGA uses the fitness value of a couple. Once a couple produces a single offspring with high fitness value, its fitness value is reduced to lower the probability of pregnancy and thus, insure the diversity of the new generation. The idea of inter-chromosomal dominance has been introduced to prevent inheritance of undesired characteristics from the parent of low fitness value. NGA has been used to solve the n -job, m -machine, lot-streaming flow shop problem with equal-size sublots with criterion of the total stretch. Extensive computational experiments have been conducted to compare the performance of NGA with that of GA. These results show that the average improvement of NGA over GA is 11.32% for medium and large scale problems.

References

- [1] Bender, M.A., Muthukrishnan, S., and Rajaraman, R., Approximation algorithms for average stretch scheduling. *Journal of Scheduling*, 2004, Vol. 7, p 195-222.
- [2] Buscher, U. and Shen, L., An integrated Tabu search algorithm for the lot streaming problem in job-shops. *European Journal of Operational Research*, 2009, Vol. 199, No. 1, p 385-399.
- [3] Chan, W.-T., Lan, T.-W., Liu, K.-S., and Wong, P.W.H. New resource augmentation analysis of the total stretch of SRPT and SJF in multiprocessor scheduling. *Theoretical Computer Science*, 2006, Vol. 359, p 430-439.
- [4] Chang, J.H. and Chiu, H.N., A comprehensive review of lot streaming. *International Journal of Production Research*, 2005, Vol. 43, No. 8, p 1515-1536.
- [5] Cheng, M., Mukherjee, H.J., and Sarin, S.C., A review of lot streaming. *International Journal of Production Research*, 2013, Vol. 51, No. 23-24, p 7023-7046.
- [6] Defersha, F.M. and Chen, M., Jobshop lot streaming with routing flexibility, sequence-dependent setups, machine release dates and lag time. *International Journal of Production Research*, 2012, Vol. 8, No. 15, p 2331-2352.
- [7] Dreoj, J., Petrowski, A., Siarry, P., and Taillard, E., Metaheuristics for Hard Optimization : Methods and Case Studies. New York, Springer, 2005.
- [8] Goldberg, D.E. Genetic algorithms in search, optimization, and machine learning. New York, Addison-Wesley, 1989.
- [9] Hall, N.G., Laporte, G., Selvarajah, E., and Sriskandrajah, C., Scheduling and lot streaming in two-machine open shops with no-wait in process. *Naval Research Logistics*, 2005, Vol. 52, p 261-275.
- [10] Kalir, A.A. and Sarin, S.C., Evaluation of the potential benefits of lot streaming in flow-shop systems. *International Journal of Production Economics*, 2000, Vol. 66, No. 2, p 131-142.
- [11] Liepins, G.E. and Hilliard, M.R., Genetic algorithms : Foundation and applications. *Annals of Operations Research*, 1989, Vol. 21, No. 1-4, p 31-58.
- [12] Low, C., Hsu, C.-M., and Huang, K.-I., Benefits of lot splitting in job-shop scheduling. *International Journal of Advanced Manufacturing Technology*, 2004, Vol. 24, p 773-780.
- [13] Marimuthu, S., Ponnambalam, S.G., and Jawahar, N., Evolutionary algorithms for scheduling m-machine flow shop with lot streaming. *Robotics and Computer-Integrated Manufacturing*, 2008, Vol. 24, No. 1, p 125-139.
- [14] Muthukrishnan, S., Rajaraman, R., Shaheen, A., and Gehrke, J.F., Online scheduling to minimize average stretch. *Siam Journal on Computing*, 2005, Vol. 34, No. 2, p 433-452.
- [15] Pan, Q.-K., Tasgetiren, M.F., Suganthan, P.N., and Chua T.J., A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, 2011, Vol. 183, p 2455-2468.
- [16] Pinedo, M.L., Scheduling : Theory, Algorithms, and Systems (4th Ed.). New York, Springer, 2012.
- [17] Tseng, C.-T. and Liao, C.-J., A discrete particle swarm optimization for lot-streaming flow-shop scheduling problem. *European Journal of Operational Research*, 2008, Vol. 191, No. 1, p 360-373.
- [18] Zhang, W., Yin, C., Liu, J., and Linn, R.J., Multi-job lot streaming to minimize the mean completion time in m-1 hybrid flowshops. *International Journal of Production Economics*, 2005, Vol. 96, p 189-200.