Diversity oriented multi-objective island based genetic algorithm for flexible job shop scheduling considering setup operator regulation

Masahiro Nagao（nagao@urban.env.nagoya-u.ac.jp）
Takuya Sugimoto, Yasuhiko Morinaga, and Mitsuru Sano
〔Nagoya University〕

段取り作業者を考慮した生産スケジューリングへの多様性指向型多目的 GA の適用

長尾 征洋、杉本 拓弥、森永 泰彦、佐野 充
名古屋大学 大学院環境学研究科

要約

製品生産現場における生産スケジュールの定式化として、フレキシブルジョブスケジューリング問題が研究の対象とされている。各ジョブショップにリソースとして製品（群）・機械・オペレータが割り当てられる研究例は多いが、段取り作業および段取り従事者に注目した研究は見られない。多品種少量生産の現場では、段取り作業者資源の効率的な配分が生産性の向上につながる。一方、人的な資源であるために、従事者の満足度は軽視できない。本研究では、段取り作業者を取り入れたフレキシブルジョブショップ問題を取り上げ、多様性指向型多目的遺伝的アルゴリズムによる解法を提案する。納期遅れ指数と段取り作業者の負荷指数を評価関数とすることで、顧客と作業従事者の視点に立った最適化が可能になった。提案手法により短時間で生産スケジュールを作成することができ、実生産データを用いた場合においても、適用可能なスケジュールを導出することができた。

Key words

flexible job shop scheduling problem, setup operator, genetic algorithm, pareto optimal, multiogjective

1. Introduction

Recently, demands for products with variety and options are increasing in the field of parts production. It is necessary to produce not only for mass production but also for small-lot production, which entails unpredictable changes in the quantity of orders and other parameters. Therefore, a new production scheduling systems corresponding to small-volume and wide variety products is expected for use in place of traditional mass production scheduling.

The job-shop scheduling problem (JSP) has received considerable attention in the production scheduling field. JSP, one of the hardest combinatorial optimization problems (Jain & Meeran, 1999), has provided proof that it is nondeterministic polynomial time (NP) hard for $m > 2$. Therefore, no optimal solution can be computed in polynomial time (Garey, Johnson, & Sethi, 1976). Although JSP has the limitation that the route of every job is given and that every operation of a job is assigned a fixed machine, the definition of JSP is extended by allowing that each operation can be processed by any machine. This complex JSP is known as the flexible job-shop problem (FJSP), which proves to be NP-hard when two jobs are allocated to unrelated machines (Mati & Xie, 2004). FJSP is classifiable into total flexibility JSP (tFJSP) and partial flexibility JSP (pFJSP) (Brucker & Schlie, 1990; Brandimarte & Calderini, 1995). The pFJSP can be regarded as tFJSP by assigning a penalty when the taboo or unprocessable operation is selected. Therefore, reports on tFJSP are increasing in the field of scheduling problems.

Because the FJSP must assign every operation to every appropriate machine and sequence the operations on each machine, an efficient algorithm for producing a rational schedule has been explored in recent years. Brucker et al. (1990) first addressed FJSP and proposed a polynomial algorithm for the FJSP with two jobs. Brandimarte et al. (1995) were the first to apply hierarchical approaches to decompose FJSP to reduce the complexity by assigning a sub-problem and sequencing problem. The complexity attributable to the flexibility creates difficulty in finding potential optimal solutions.

Metaheuristic approaches are proposed to solve FJSP and to improve the algorithm performance, because metaheuristics with effective problem mapping and solution generation ability make it possible to attain the optimized solutions. Mastrolilli & Gambarardella (2000) proposed neighborhood functions for metaheuristics. Ong, Tay, & Kwon (2005) applied the clonal selection principle of the human immune system to solve FJSP with re-circulation. Mesghouni, Hammadi, & Borne (1998) applied genetic algorithm to FJSP and developed new genetic operators combining the assignment and scheduling problem successfully. Jia, Fuh, Nee, & Zhang (2007) proposed a genetic algorithm (GA) integrated with a Gantt chart (GC) to solve job shop scheduling with a distributed manufacturing environment. They concluded that the application of the GC enabled fast chromosome evaluation. Kacem, Hammadi, & Borne (2002a) proposed a genetic algorithm controlled by approach by localization (AL) and an assignment model to both t-FJSP and p-FJSP for solving single-objective and multi-objective optimization.

Although empirical research on FJSP has specifically addressed single-objective problems, multiple objectives should be examined simultaneously in the real-world production situation. The possible objectives for scheduling such as make span, total workload, maximum workload, idle time, set-up time, tardiness, maximum tardiness, completion time, number of late jobs, and others often mutually conflict (Wang, Gao, Zhang, & Shao, 2010) because evaluation of the objective might depend on the departmentalization of the decision makers, whereby each decision maker tries to optimize their given criterion. Multi-objective FJSP is closer to a realistic production schedule and must be solved effectively using metaheuristic approaches.

The aggregation approaches using metaheuristics often combine all the objective functions into a single function. For example, when it is necessary to minimize every objective function, the objective functions are summarized into a single function $\sum_{i=1}^{k} w_i F_i$, where wi is the weight of the objective F_i. Loukil et al. proposed simulation annealing for solving multi-objective FJSP in a Tunisian firm. A scalarizing function is applied to aggregate multiple performances into a unique one (Coello, 2005). Low, Yip, & Wu (2006) proposed hybrid heuristics combining two local search methods—simulated annealing (SA) and tabu search (TS)—for solving flexible manufacturing system. The SA/TS hybrid algorithm showed the best performance for large problem size (job size $N \geq 10$). They described that the weight of objectives might affect the schedule performance. Li, Pan, & Liang (2010) developed a TS algorithm with an effective neighborhood structure for solving the multi-objective FJSP. A hybrid tabu search algorithm demonstrated its superiority over other hybrid approaches such as AL+CGA, particle swarm optimization (PSO)+SA, PSO+TS, and Xing's algorithm (Xing, Chen, & Yang, 2008). Azardoost & Imanipour (2011) proposed a hybrid metaheuristic algorithm based on TS, SA, and GAs with suitable parameters for solving FJSP. Grobler, Engelbrecht, Kok, & Yadavalli (2010) proposed penalty-based PSO, priority-based PSO, random keys PSO, and rule-based PSO. Their results indicate that priority-based representation showed significant improvement in performance, which led to the development of a priority-based differential evolution algorithm. Zhang, Shao, Li, & Gao (2009) introduced a hybridized PSO approach for multi-objective FJSP. Liu, Abraham, Choi, & Moon (2006) developed variable neighborhood PSO. The dynamic weighted aggregation is used to adapt dynamically during optimization calculation. Xing, Chen, & Yang (2009) designed a simulation model to address multi-objective FJSP and evaluated it using the ant colony optimization (ACO) algorithm. They also proposed a double-layer ACO algorithm consisting of an upper layer to assign operations on machines and a lower layer to schedule operations on each machine. Tay & Ho (2008) evaluated suitable parameters and operator spaces for evolving composite dispatching rules using genetic programming. Saad, Hammadi, Benrejeb, & Borne (2008) proposed a GA using a Choquet integral as a tool for dealing with multiple criteria decision-making. Dalfard & Mohammadi (2012) applied a hybrid GA and a SA for solving multi-objective FJSP. Although the aggregation approaches are often applied to solve the multi-objective FJSP, they are not always efficient for exploring non-convex objective space (Coello, 2005). Intensive study like that described above reflects the potential interest in multi-objective FJSP by metaheuristics.

Recently, Pareto-based approaches are gathering attention for determination of the optimal solutions for multi-objective FJSP by application of a dominance concept, where a set of all non-dominated alternative solutions is defined as a Pareto-optimal front. However, reports of research on the multi-objective FJSP with Pareto-based approach are scant compared to those of the hierarchical or non-Pareto approaches listed above. Kacem, Hammadi, & Borne (2002b) proposed a Pareto approach based on evolutionary algorithms improved by hybridization with fuzzy logic to solve the multi-objective FJSP. Ho and Tay (2008) introduced guided local search instead of random local search and combined it with an evolutionary algorithm. Many of the obtained Pareto-optimal solutions showed superiority in quality, range, and computation performance. Wang et al. (2010) proposed a multi-objective GA with immune and entropy principle. Decrease in selection pressure of similar individuals by combining the immune and entropy principle made the proposed algorithm effective as a multi-objective approach. Frutos, Olivera, & Tohmé (2010) studied a memetic algorithm that combines NSGA-II with SA as a guided local search. The proposed algorithm provided an efficient calculation of the objective functions and varied solution at the same time. Li, Pan, & Gao (2011) developed a Pareto-based discrete artificial bee colony algorithm (P-DABC) and introduced improved crossover. The P-DABC can obtain both superior solutions and richer non-dominated solutions than the other compared algorithm can. Wang, Zhou, Xu, & Liu (2012) proposed an enhanced Pareto-based artificial bee colony algorithm (EPABC) to solve the multi-objective FJSP. Through improvement of the initialization scheme, exploitation search procedures, crossover operators for machine assignment and operation sequence, local search based on critical path, recombination, and selection strategy, the proposed EPABC showed better performance than the other algorithm including P-DABC. Moslehi & Mahnam (2011) proposed a new approach based on a hybridization of the particle swarm and local search algorithm to both weighted summation of objectives and Pareto approach. Even large problems, the proposed algorithm generated better solutions than benchmark. Li, Pan, & Xie (2012) proposed a hybrid shuffled frog-leaping algorithm and introduced a well-designed crossover operator in the proposed algorithm. The designed neighborhood structures realized a promising search space. The Pareto approach to multi-objective FJSP is only rarely discussed in the literature because of the complexity of multi-objective FJSP.

Which indexes should be optimized using the Pareto approach depends on the decision-maker. The total tardiness of each product and the difference of the setup number among setup operators are regarded as objectives. We consider a human resource, a setup operator, who prepares the material or changes the tools needed for proceeding to the next process because, in a company producing a variety of products, the load attributable to the setup process is not negligible and is gaining attention because of its broad applicability in fields such as precision equipment manufacturing (Allahverdi, Ng, Cheng, & Kovalyov, 2008; Roshanaei, Naderi, Jolai, & Khalili 2009). They are optimized using the Pareto-based method in this study. This report is the first describing installation of setup operators for the restriction of human resources and an objective function that evaluates the fairness of the labor load.

We describe herein that the setup time of a job by a setup operator depends on the next machine by which the job should be processed. Intensive studies of the sequence-dependent setup time (SDST) have been reported, where the setup is strongly dependent on the immediately preceding process on the same processor (Huang, Süer, & Urs, 2012; Gröflin, Pham, & Bürgy 2011; Roshanaei, Balagh, Esfahani, & Vahdani 2010; Kurihara, Li, Nishiuchi, & Masuda 2009). In a real manufacturing company, although the SDST is important to minimize inventory costs, it is difficult to determine SDST exactly from theoretical or empirical data because $O(n^2)$ of setup times should be determined when n products are processed in one machine. The separability and independence of setup time were discussed in detail for the $M \times N$ flowshop problem and sequence-independent setup time (SIST) was investigated for its impact on competing flowshop system performance measures (Stafford & Tseng, 2006). The setup time in this report is regarded as dependent solely on the next processing machine: the setup time of a job on a machine is regarded as constant.

No published work describes multi-objective FJSP considering setup operators using a distributed (or island based) genetic algorithm. A distributed approach using islands mode is applied to maintain the diversity of chromosomes (Tanese, 1989; Park, Choi, & Kim, 2003). The search direction toward two objectives can be controlled using a weight coefficient which gives variation in the evaluation, even for the same chromosome with respect to the preference of each island.

As described in this paper, we propose a distributed multi-objective genetic algorithm with variable congestion to solve the FJSP considering the setup operator constraint. To make a valid comparison of the evaluation values, which have different orders of magnitude, the degree of congestion was proposed by normalizing the evaluation values for the calculation of Euclidean distance as the congestion degree. The performance of the proposed algorithm was discussed with respect to the diversity of the solutions by the introduction of the island-based distributed method and the diversity-oriented evaluation of the chromosomes and the local search ability by the emigration. The proposed algorithm is examined for validation in two test instances: one has moderate problem complexity to obtain the exact solutions; the other has complexity that reflects the production scene in the realistic enterprise. The originality of this research is summarized as follows:

1. FJSP considering setup operators as a second resource and constraints
2. Order-based representation corresponding to three indexes (e.g., (machine, process time, setup operator))
3. Introducing idle time in the gene representation for multi-objective FJSP.
4. Diversity-oriented approach in island-based GA (IGA) using a weight coefficient for the evaluation function in each island.
5. The congestion degree is calculated according to normalized evaluation values

The remainder of this paper is organized as follows. Section 2 introduces the problem formulation and proposed GA. Computational results for the test instance with small problem scale are reported in Section 3. The applicability of the proposed algorithm to the example of the realistic enterprise and comparison with other methods are discussed in section 4. Finally, some conclusions inferred from our work are presented in the last section.

2. Problem formation

The flexible job-shop scheduling problem (FJSP) is described as follows: m machines are given for n jobs. Job J_i must be processed in o_{ki} operations. Each operation O_{ij} should be assigned to one of the capable machines of the jth operations for job J_i. The processing time of the jth operations for job J_i on the machine k is denoted as P_{ijk}. Both the assignment of machines and the sequence of operations should be determined on the FJSP. Some constraints should be met in the FJSP.

1. Each machine can process only one operation at a time.
2. Each machine starts at time 0.
3. Each machine can process without maintenance but a setup can be skipped when the following process is the same lot of the job or similar product group of the job.
4. Each operation can be processed only on one machine at a time. (One lot cannot be divided into other lots to proceed with multiple machines.)
5. Each operation cannot be processed without completion of its preceding process.

6. No interruption of each operation is allowed when the operation is proceeding on the machine.
7. No sequence restriction exists among operations for different jobs.

In this study, two objectives are regarded as shown below.

$$\min\ L = \sum_{k=1}^{K} w_k T_k \tag{1}$$

$$Q_{\max} = \max\{Q_u | u = 1,2,\cdots U\} \tag{2}$$

$$Q_{\min} = \min\{Q_u | u = 1,2,\cdots U\} \tag{3}$$

$$\min\ R = Q_{\max} - Q_{\min} \tag{4}$$

In those equations, K stands for the total number of jobs. L is the summation of weighted tardiness of all jobs. w_k signifies the weight coefficient for kth jobs, which determines the priority of the jobs and T_k denotes the tardiness of each job on a day-to-day basis. R represents the setup operators' load leveling index. Maximum and minimum values of the setup number among setup operators are denoted respectively as Q_{max} and Q_{min}. Q_u is the sum setup number of setup operator u. When a setup operator is assigned to more than one machine during the same time span, a penalty value is imposed to R (setup operator regulation).

The encoding scheme plays an important role in accomplishing effective searching in the objective space. Among several representation methods, order-based representation is often used for a scheduling problem. The relative orders of the jobs on each machine are shown in the permutation. By scanning the permutation from left to right, the kth appearance means the kth possible operation of the machine if the kth operation is ready to proceed. In this representation method, because the permutation of the jobs on each machine is connected one-dimensionally, each job number appears J_k times or more in the permutation. The setup operator is assigned according to the machines by reference to the skill map. Here we presume that the machine is linked to one or more setup operators who can accomplish the setup in constant time. We consider the two jobs on two machines scheduling with two setup operators.

The information of the job is provided as a set of the operations as shown in Table 1. Each cell consists of three components, (machine ID, process time, setup operator ID), where the first term represents the ID of machine M_m, which performs the operation. The second one represents the time necessary to complete the operation. The last one represents the ID of the setup operator. For J_2 in Table 1, two possible setup operators exist for machine M_2 for the first operation and also two possible machines for second operation. The gene which appeared in the chromosome in Figure 1 is represented according to following formula.

$$k + 2(h - 1) \tag{5}$$

M_1	1	2	-1	-1				
M_2	1	4	3	3	6	6	-1	-1

Figure 1: Encoded chromosome

Therein, k is the number of jobs; h is the maximum number of combinations of possible machines and setup operators.

A scheduling solution using a Gantt chart is applied to evaluate each chromosome. When the first gene of machine M_1 is read, it occupies the schedule for corresponding time together with setup time. It next moves on to the first gene of machine M_2 and does the same procedure. After completion of subsequent machines, the reading sequence moves on to the second gene of each machine. An idle time is inserted if the sequence reaches -1. Figure 2 depicts an example of the chromosome. Figure 3 is a Gantt chart obtained from the chromosome presented in Figure 2.

M_1	**-1**	**1**	**2**	-1				
M_2	4	6	1	3	6	4	-1	-1

Figure 2: Example of encoded chromosome

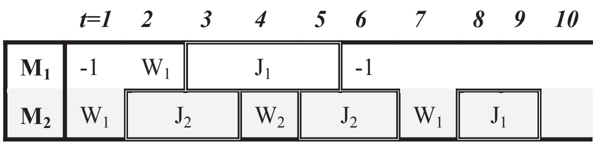

Figure 3: Gantt chart produced from the chromosome in Figure 2

Table 1: Example of FJSP with two jobs and two setup operators and its gene representation

k	Job	h	1st operation	2nd operation	Representation in genes k + 2(h - 1)
1	J1	1	(M1,3,W1)	(M2,2,W1)	1
		2		(M2,2,W2)	3
2	J2	1	(M2,2,W1)	(M2,2,W1)	2
		2	(M2,2,W2)	(M2,2,W2)	4
		3		(M1,2,W1)	6

3. Multi-objective optimization and Pareto optimality approach

Our proposed algorithm is based on the NSGA-II, a Non-dominated sorting genetic algorithm proposed by Deb et al. for the optimization calculation (Deb, Pratap, Agarwal, & Meyarivan, 2002). It also introduces the parallel mode using multiple-island model proposed by Tanese (1987) and Cohoon, Hedge, Martin, & Richards (1987). NSGA-II is a well-known multi-objective evolutionary algorithms with potential to be improved to obtain a widely spread non-dominated frontier.

Let us consider a general multi-objective minimization problem in the following form.

$$\min \; y = f(x) = (f_1(x), f_2(x), f_3(x), \ldots, f_q(x)) \tag{6}$$

Therein, $x \in R^p$, and $y \in R^q$, p is the dimension of the variable x, and q is the number of sub-objectives (Tavakkoli-Moghaddam, Azarkish, & Sadeghnejad-Barkousaraie, 2011). To address multi-objective optimization, we use the Pareto-optimal concept. Pareto-optimal is defined as a feasible solution that is not dominated by any other solution in the search space. The Pareto-optimal set is the collection of all Pareto-optimal solutions. Their corresponding images in the objective space are called the Pareto-optimal frontier.

Solution a is said dominate solution b if and only if the following holds.

$$f_i(a) \le f_i(b), \qquad \forall i \in \{1,2,\ldots,q\} \tag{7}$$

$$f_i(a) < f_i(b), \qquad \exists i \in \{1,2,\ldots,q\} \tag{8}$$

All chromosomes are sorted and ranked: for example, every non-dominated solution is assigned as a domination rank 1. All non-dominated solutions of the rest have a domination rank 2, and so on. These domination ranks of all solutions are used to determine the parents who give the offspring for the next generation. NSGA-II adopts an elitist strategy joint with an explicit mechanism to ensure the diversity of exploring space. The algorithm starts by creating an initial population at random. A population Q_t obtains from a parent population P_t. R_t (of size $2N$) is obtained from the combination of Q_t and P_t. The members of R_t are classified into several Pareto frontiers using non-dominated alternatives. According to this idea, the isolated individuals are selected preferentially to produce offspring. A non-dominance ranking r_i as well as a congestion degree, which represents the total distance among other individuals belonging the same Pareto frontier, is used for the selection.

The congestion degree is defined as the crowded state of an individual in the Pareto frontier. The Euclidean distance between individuals a and b who belong to the same Pareto frontier is calculated as follows using normalized coordinates.

$$nL_a = \frac{L_a - L_{ave}}{L_{ave}} \tag{9}$$

$$nR_a = \frac{R_a - R_{ave}}{R_{ave}} \tag{10}$$

$$distance_{\mathrm{a,b}} = \sqrt{(nL_a - nL_b)^2 + (nR_a - nR_b)^2} \tag{11}$$

In those equations, nL_a and nR_a respectively signify the normalized evaluated values with respect to L_a and R_a of individual a. L_{ave} and R_{ave} respectively denote the averages of L and R values of individuals in the same Pareto frontier. $distance_{a,b}$ represents the Euclidean distance separating individuals a and b. The congestion degree is calculated as

$$congestion_{\mathrm{a}} = \sum_{b=1}^{F_f - 1} \sqrt{q(nL_b - nL_{b+1})^2 + (1-q)(nR_b - nR_{b+1})^2} \tag{12}$$

where $congenstion_a$ is the congestion degree of individual a. F_f stands for the number of individuals in the Pareto frontier which individual **a** is involved. q denotes the weight coefficient to give the difference in the evaluation function between the island in the proposed algorithm.

In IGA, a group of chromosomes is divided into several subgroups called islands. For each island, a set of GA operation is applied for seeking optimal solutions separately. Several GA operators (i.e., size, crossover rate, mutation rate) are changeable for each island. However, each operator is fixed in every island to avoid complexity. Each evolution calculation is performed on the same single processor, i.e., our IGA is not strictly the same with Parallel GA, where evolution of chromosome takes place by multiple processors (Park et al., 2003).

In IGA, some chromosomes emigrate to other islands at fixed intervals. Top 5 chromosomes of each island with respect to the congestion degree are selected and listed in the emigration set (ES). Five randomly selected chromosomes on the ES are returned to every island to mate and create offspring. The emigration operation described above is performed every 5,000 generations in this study.

When the congestion degree is calculated to give the ranking of the chromosomes in the same Pareto frontier, the weight coefficient q is introduced into eq. 12. q is changed from 0 to 1 with constant intervals. When q is close to 1, it is considered that the congestion degree is weighted toward L.

To evaluate the performance of the proposed GA, Single-objective GA (SOGA) is defined as shown below.

1. SOGA optimizes only one objective, L.
2. Congestion degrees and diversity-oriented island mode are not included in SOGA because SOGA need not adopt the Pareto approach.

The proposed algorithm starts with the initialization of population. Each gene of a chromosome is set at random from the possible number in the permutation. Random population will allow the search process in a wide exploration space.

Crossover operators are used to generate the offspring and applied to pairs of chromosomes (called parents). It should be kept in mind that the variety of solutions should be preserved while allowing the exchange of the order representation. In the proposed algorithm, crossover methods of two kinds are applied.

Two crossover points are selected randomly as a crossing region. Crossover is performed in the crossing region to obtain the transitional offspring. To avoid the replicated allele found in the transitional offspring, it is filled by cross-referencing with the parent of the alternate chromosome (Goldberg & Lingle, 1985). The procedure, partially matched crossover, is presented in Figure 4.

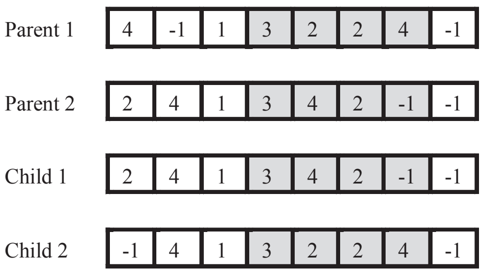

Figure 4: Example of partially matched crossover operation

Segment-based crossover operation is implemented by swapping two segments between two parent chromosomes. Chromosomes are segmentalized by machines. A stochastically selected segment of parent 1 and the same segment of parent 2 are exchanged to create two new offspring. The two selected segments represent the same machine line, thereby insuring the feasibility of the offspring in the chromosome representation. The procedure is presented in Figure 5.

Mutation is often adopted to prevent a loss of diversity in the chromosomes. Randomized exchange of alleles in the chromosomes is applied as the mutation. This operation occurs within

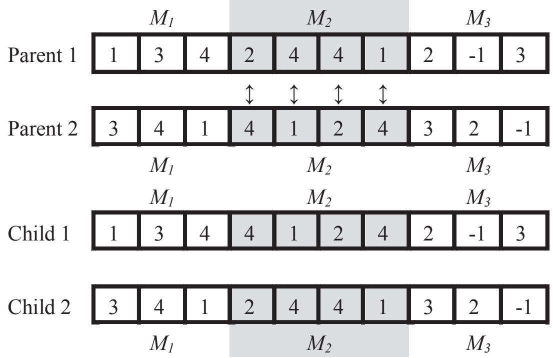

Figure 5: Example of segment-based crossover operation

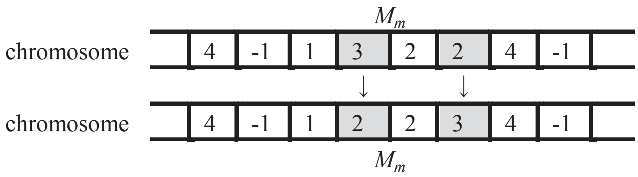

Figure 6: Example of mutation operation

single chromosome with a given probability. The procedure is presented in Figure 6.

In the test instance 1, we consider the schedule with 5 jobs × 12 machines and each operation is started by the three setup operators. Details of the assignment of the jobs and machines on the operation are presented in Table 2.

The proposed algorithm was implemented in C++ on a Core(TM) i7-3770 3.40 GHz CPU with a Windows 7 64-bit operation system. For each trial, 10 independent runs were performed.

Number of chromosomes, *R*	144
Scheduling period, *G*	7 days
Termination condition	30 s
Weight coefficient, *q*	0.5 (constant)
Emigration	N/A

Table 2: Example of FJSP with five jobs and three setup operators and gene representation

k	Job	h	Lot	time left	1st operation	2nd operation	3rd operation	4th operation	5th operation	Representation in genes
1	J_1	1	20	27	$(M_3, 4, W_2)$	$(M_8, 0.17, *)$	$(M_7, 2, W_2)$	$(M_1, 3, W_1)$	$(M_5, 1, W_2)$	1
		2							$(M_5, 1, W_3)$	6
		3							$(M_3, 2, W_2)$	11
2	J_2	1	150	29	$(M_3, 2, W_2)$	$(M_8, 0.17, *)$	$(M_7, 1, W_2)$	$(M_5, 1, W_2)$	$(M_{10}, 1, *)$	2
		2						$(M_5, 1, W_3)$		7
3	J_3	1	10	17	$(M_2, 4, W_1)$	$(M_4, 4, W_2)$	$(M_9, 0.17, *)$	$(M_7, 2, W_1)$	$(M_{11}, 42, *)$	3
		2					$(M_{12}, 42, *)$	$(M_6, 2, W_2)$	$(M_{12}, 42, *)$	8
		3						$(M_7, 2, W_2)$		13
4	J_4	1	15	25	$(M_3, 4, W_2)$	$(M_8, 0.17, *)$	$(M_7, 2, W_2)$	$(M_1, 3, W_1)$	$(M_3, 2, W_2)$	4
		2							$(M_5, 1, W_2)$	9
		3							$(M_5, 1, W_3)$	14
5	J_5	1	30	11	$(M_2, 3, W_1)$	$(M_4, 2, W_2)$				5

Note: * denotes that the operation is processed without setup, where the operation is free from setup operator regulation and where it does not count the setup time.

4. Results and discussion

4.1 Test instance 1: application to the simulated company with small scale production

The problem size is moderated to obtain the exact solution. Two exact solutions were developed using the enumeration method. It took 28.5 s, on average, to obtain two exact solutions. From the obtained Gantt chart, the summation of weighted tardiness of all jobs, L, is 0 and the set-upper load leveling index, R, is 6 for #1 solution and L is 30 and R is 4 for #2 solution. A Gantt chart of one of the two (#1 solution; $L = 0$, $R = 6$) is portrayed in Figure 7.

When the proposed GA is applied to this test instance, it took 3.5 s in average to obtain the two exact solutions. The transition of the solution candidates with the Pareto frontier is presented in Figure 8. Both the x-axis and y-axis are displayed before normalization of the evaluated values. From this result, the proposed genetic algorithm is suggested as applicable to FJSP, which is down-seized from the realistic scheduling problem as well as the enumeration method.

To set and optimize the genetic parameters in the proposed algorithm, we define the rate of exact solution attainment, R_{exact}, and calculate this index for every combination of the parameters presented in Table 3. Because the problem size of this test instance is sufficiently small to obtain the exact solutions in the finite time, we can discuss the performance of the proposed algorithm by comparison of whether the obtained solutions are the same with the exact solutions. The definition of the rate of exact solution attainment, R_{exact}, is as follows.

Table 3: Parameters of GA

Number of islands	1, 2, 4, 8, 12, 18, 24
Cross over rate(%)	
Partially matched crossover	0, 30, 50, 70, 100
Segment-based crossover	0, 30, 50, 70, 100
Mutation rate(%)	0, 10, 20, 30, 40

$$R_{exact} = \frac{N_{exact}}{M_{trial}} \times 100 \tag{13}$$

Therein, M_{trial} is the number of independent repetition trials. N_{exact} is the number of trials which reach the exact solution.

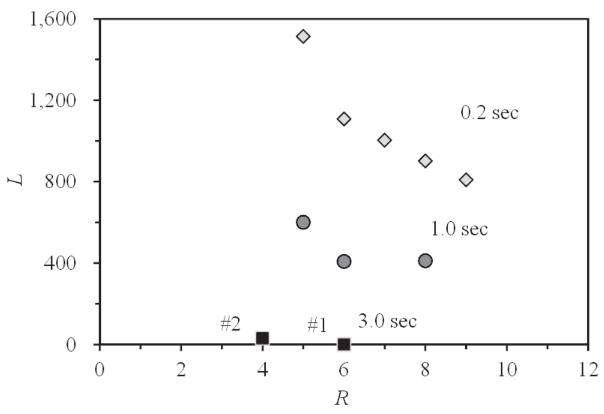

Figure 8: Time dependence of the Pareto frontiers

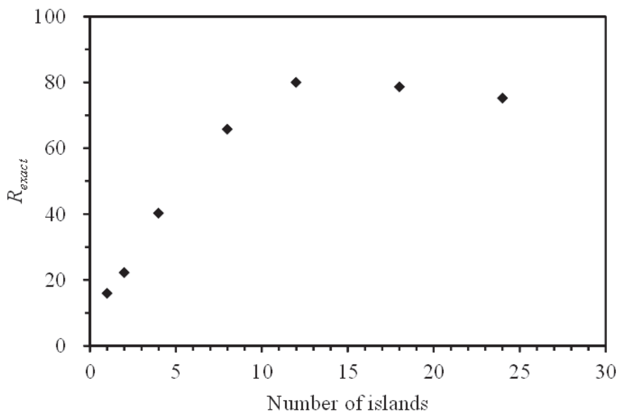

Figure 9: Dependence of R_{exact} on the number of islands

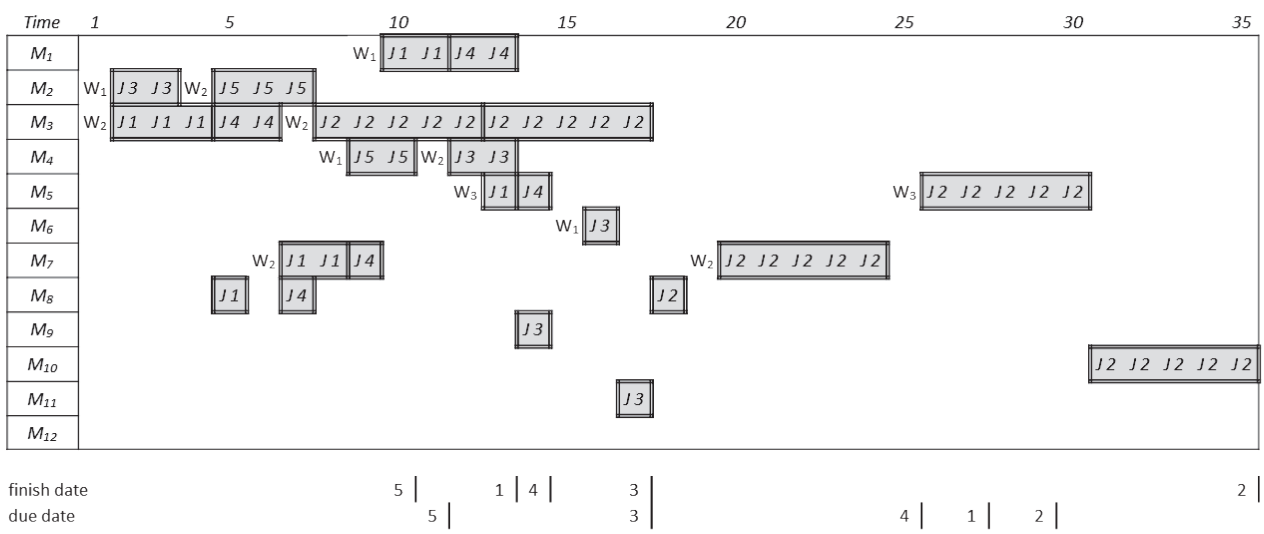

Figure 7: Gantt chart produced from one of the obtained solution #1

Dependence of the number of islands on the R_{exact} is presented in Figure 9.

When the number of islands is one, the proposed algorithm is equivalent to the simple multi-objective GA. The obtained R_{exact} is less than 20%, which suggests that once the group of chromosomes has reached the local solution, where most members of the group have a similar representation in the chromosome description, it is difficult to be transferred or come out from the local space to the other possible spaces through genetic operations such as crossovers and mutation.

The R_{exact} increases concomitantly with increasing number of the islands and reaches a maximum at around 12 islands. From this result, diversity of the search space was introduced by the parallel operation using the island-based model. On the other hand, R_{exact} started to decrease when the number of islands exceeded 12 because the local search is no longer dominant when the number of chromosomes is insufficient for reaching the local solution. Additionally, we should address that the effect of the number of islands, total chromosomes and chromosomes in each island on the performance will depend on the problem complexity.

The average R_{exact} values obtained when the mutation rate, crossover rates are changed are presented in Table 4. For example, the average R_{exact} when the segment-based crossover rate is 0 %, is calculated as follows: For the summation of R_{exact} values for every 10 trials when the quantities of islands are changed from 1 to 24, the mutation rates are changed from 0 to 40 % and partially matched crossover rates are changed from 0 to 100 % is divided by 1,750: the total number of trials.

When the mutation rate is 0, the obtained R_{exact} is extremely low: 0.2%. Even if two crossover rates are changed to higher rates, the maximum R_{exact} is around 75%, which suggests that in the remaining 20 % trials, the crossover operation cannot produce offspring effectively without mutation. In other words, the chromosome diversity is insufficient to reach the optimal solution when the mutation does not work. Regarding the results presented above, the mutation rate, crossover rate for partially matched crossover and segment-based crossover are fixed respectively as 20 %, 30 %, and 50 % for the following numerical examinations.

4.2 Test instance 2: Application to a real company producing pulleys

In the next test instance, we consider the schedule with 78 jobs × 46 machines. Each operation is started using a setup operation. Eight setup operators exist. This setting reflects the realistic production planning of June 4, 2011 of the model enterprise where producing pulleys for precision machinery and manufacturing machine.

Number of chromosomes, R	144
Scheduling period, G	19 days
Termination condition	60,000 s
Weight coefficient, q	0-1.0
Emigration	5,000

To compare the effectiveness of the introduction of the emigration operation and the diversity-oriented island, four cases are regarded as shown in Table 5.

Figures 10 shows the Pareto frontiers of four cases in 500, 1,000, 2,500, 5,000, 7,500, 15,000, 30,000, and 60,000 s. The

Table 4: R_{exact} values at various mutation rates and crossover rates

Mutation rate (%)	Rate of exact solution attainment (%)
0	0.2
10	94.0
20	95.4
30	94.7
40	93.6

Segment-based crossover rate (%)	Rate of exact solution attainment (%)
0	75.6
30	75.4
50	75.8
70	75.4
100	75.6

Partially-matched crossover rate (%)	Rate of exact solution attainment (%)
0	75.8
30	75.0
50	75.8
70	74.3
100	76.9

Table 5: Proposed GAs with/without emigration and diversity-oriented evaluation

	Description	Emigration	Diversity-oriented evaluation
1	GA1	Not available	Not available
2	GA2	Available	Not available
3	GA3	Not available	Available
4	GA4	Available	Available

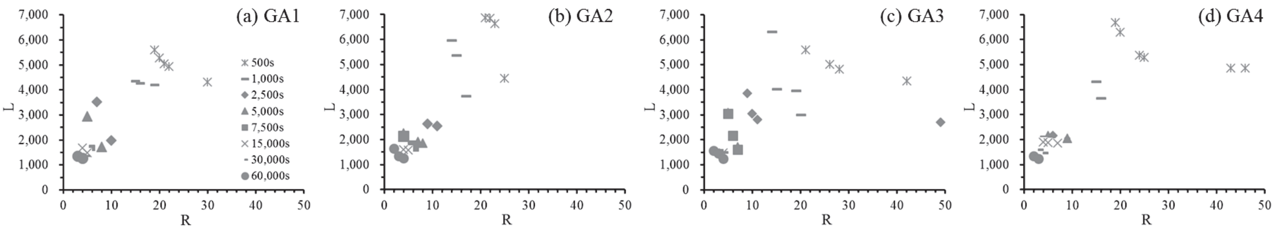

Figure 10: Time changes of the Pareto frontiers for (a) GA1, (b) GA2, (c) GA3 and (d) GA4

availability of the emigration and diversity-oriented evaluation are discussed by comparing four cases. When the result of GA2 is compared with GA1, the trajectory of the Pareto frontiers gradually became narrow and eventually converged to a few local solutions. The better chromosomes will be exchanged among islands and start to search with self-referencing in new island. The performance on local search was improved by introducing emigration. When the result of GA3 is compared with GA1, the distance between the solutions in each Pareto frontier is long in every generation presented in Figure 10.

Although the large search space was maintained in GA3, the solutions in the final Pareto frontier were inferior to that in GA1. By introducing the weighing coefficient, q, the search direction of each island was maintained through the generations. When the emigration and diversity-oriented evaluation method were introduced simultaneously (GA4), the obtained results converged on two near-optimal solutions, which are the most Pareto effective among the four tested GAs. Although the trajectory of the Pareto frontiers was kept wide before emigration at early generations, the local search worked well to identify a local solution at each generation which emigration was taken place. The results shown above and the scheme of the emigration, local search function is strengthened by the introduction of the superior chromosomes to the other islands. Together with the wide search space introduced by the diversity-oriented evaluation, the search performance of GA4 was improved to obtain better Pareto solutions than the others.

Next, the number of islands is changed from 8 to 36 to ascertain whether they can affect the efficiency of the proposed algorithm. The algorithm is based on the GA4. In each case, the calculation procedure is run independently three times. The Relative Percentage Deviation, *RPD*, the deviation of the solution versus the best solution obtained by each method is introduced to evaluate the performance of them with several island numbers. *RPD* is calculated as follows.

$$RPD = \frac{Value - Value_{(Best)}}{Value_{(Best)}} \times 100 \qquad (14)$$

The obtained *RPD* values are presented in Table 6.

Table 6: RPD_L and RPD_R values according to the number of islands

Number of islands	RPD_L	RPD_R
8	12.9	11.1
12	6.3	16.7
18	8.8	16.7
24	8.7	16.7
36	51.1	86.7

The *RPD* values related to L, RPD_L, were lowest when the number of islands was 12. However, they showed a tendency to decrease concomitantly with increasing the number of islands. The *RPD* values related to R, RPD_R, increased according to the number of islands. From this result with 12 islands, the obtained objective values had a narrower gap separating them than the other methods. In other words, the results obtained by GA4 with 12 islands are regarded as fair and acceptable because of the low *RPD* values.

Solutions obtained using the proposed algorithm compared with those obtained using the early due date (EDD) method and SOGA are presented in Table 7. Here, SOGA means the simple classical GA optimizing the L value. SOGA does not include the concept of islands but uses similar chromosome representations.

In the EDD, because the proceeding sequence is arranged with respect to the rest time for the due date, the R value is not optimized during the process. Therefore, the R value of EDD was the worst among all methods. The SSGA is slightly improved on both L and R values compared to EDD. The me-

Table 7: Comparison of *L* and *R* values with the results of EDD, SOGA and GA4

	EDD	SOGA	GA4	
			Solution #1	Solution #2
L	1339	1247	1349	1251
R	61	36	2	4

taheuristic approach allows exploration of the locally non-optimized but globally optimized solution space. However, the SSGA aims for optimization of only the *L* value, and the improvement in *R* value are expected to depend on the trials or the calculation time. Two Pareto solutions were obtained using the proposed GA with adjusted parameters.

5. Conclusion

A new production scheduling system corresponding to small-volume and widely diverse products is anticipated to take the place of traditional production scheduling. We propose a distributed multi-objective genetic algorithm with variable congestion to solve the FJSP considering a set-upper constraint. The multi-objective multiple-island genetic algorithm is improved by introducing the diversity of island for realizing the strengthening the local search and maintaining the search space at the same time. The proposed algorithm is examined for validation on two test instances: one has the moderate problem complexity to obtain the exact solutions. The other has complexity reflecting the production scene in a realistic enterprise. When the proposed GA is applied to the test instance with moderate size of the problem complexity, it can obtain the two exact solutions within a few seconds. Results show that the diversity of the search space is introduced by the parallel operation with 12 islands. To demonstrate the applicability of the algorithm to the production system of the realistic company, optimization calculations were applied to the realistic production planning of the model enterprise producing pulleys for precision machinery and manufacturing machines. Introduction of the diversity-oriented evaluation of each island is also expected to improve the local search of each island according to their preference.

The performance of proposed GA will depend on the problem complexity. However, it will be applicable to obtain optimal or sub-optimal solutions effectively by controlling variable numbers such as the quantities of islands and chromosomes in each island and the rates of the mutation and crossovers. This point warrants future work on more complex manufacturing systems and resource regulations such as setup operators in this work. Nonetheless, the newly developed genetic algorithm for solving FJSP with setup operators will enable us to produce several Pareto solutions and to select the best schedule according to their intention.

Acknowledgements

Detailed manufacturing data were provided by Mrs. Hiroe Yoshida at Yoshida-TK Corporation. Fruitful discussions with Mr. Norio Kawahara at the Okazaki Shinkin Bank were greatly appreciated.

References

Allahverdi, A., Ng, C. T., Cheng, T. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187, 985-1032.

Azardoost, E. B., & Imanipour, N. (2011). A hybrid algorithm for multi objective flexible job shop. scheduling problem. *Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management*. Kuala Lumpur. Malaysia.

Brandimarte, P. & Calderini, M. (1995). A hierarchical bicriterion approach to integrated process plan selection and job shop scheduling. *International Journal of Production Research*, 33, 161-181.

Brucker, P. & Schlie, R. (1990). Job-shop scheduling with multipurpose machines. *Computing*, 45, 369-375.

Cohoon, J. P., Hedge, S. U., Martin, W. N., & Richards, D. (1987). Punctuated Equilibria: a Parallel genetic Algorithm. *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, Lawrence Erlbaum Associates, 148-154.

Coello, C. A. C. (2005). Recent trends in evolutionary multiobjective optimization in Evolutionary Multiobjective Optimization. *Advanced Information and Knowledge Processing*, 7-32.

Dalfard, V.M. & Mohammadi, G. (2012). Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance constraints. *Computers & Mathematics with Applications*, 64, 2111-2117.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 182-197.

Frutos, M., Olivera, A. C., & Tohmé, F. (2010). A memetic algorithm based on a NSGAII scheme for the flexible job-shop scheduling problem. *Annals of Operations Research*, 181, 745-765.

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and job shop scheduling. *Mathematics of Operations Research*, 1, 117-129.

Goldberg, D. E. & Lingle, R. (1985). Alleles, loci, and the Traveling Salesman Problem. *Proceedings of the First International Conference on Genetic Algorithms and Their Application*, Hillsdale, 154-159.

Grobler, J., Engelbrecht, A. P., Kok, S., & Yadavalli, S. (2010). Metaheuristics for the multi-objective FJSP with sequence-dependent set-up times, auxiliary resources and machine down time. *Annals of Operations Research*, 180, 165-196.

Gröflin, H., Pham, D. N., & Bürgy, R. (2011). The flexible blocking job shop with transfer and set-up times. *Journal of Combinatorial Optimization*, 22, 121-144.

Ho, N. B. & Tay, J. C. (2008). Solving multiple-objective flexible job shop problems by evolution and local search. *IEEE Transactions on Systems Man and Cybernetics Part C*, 38, 674-685.

Huang, J., Süer, G. A., & Urs, S. B. R. (2012). Genetic algorithm for rotary machine scheduling with dependent processing times. *Journal of Intelligent Manufacturing*, 23, 1931-1948.

Jain, A. S. & Meeran, S. (1999). Deterministic job-shop scheduling: past, present and future. *European Journal of Operational Research*, 113, 390-434.

Jia, H. Z., Fuh, J. Y. H., Nee, A. Y. C., & Zhang, Y. F. (2007). Integration of genetic algorithm and Gantt chart for job shop scheduling in distributed manufacturing systems. *Computers & Industrial Engineering*, 53, 313-320.

Kacem, I., Hammadi, S., & Borne, P. (2002a). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 32, 1-13.

Kacem, I., Hammadi, S., & Borne, P. (2002b). Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. *Mathematics and Computers in Simulation*, 60, 245-276.

Kurihara, K., Li, Y. L., Nishiuchi, N., & Masuda, K. (2009). Flow shop scheduling for separation model of set-up and net process based on Branch-and-Bound method. *Computers and Industrial Engineering*, 57, 550-562.

Li, J. Q., Pan, Q. K., & Liang, Y. C. (2010). An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Computers and Industrial Engineering*, 59, 647-662.

Li, J. Q., Pan, Q. K., & Gao, K. Z. (2011). Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems. *The International Journal of Advanced Manufacturing Technology*, 55, 1159-1169.

Li, J. Q., Pan, Q. K., & Xie, S. (2012). An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems. *Applied Mathematics and Computation*, 218, 9353-9371.

Liu, H., Abraham, A., Choi, O., & Moon, S. H. (2006) Variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems. *Simulated Evolution and Learning*, 4247, 197-204.

Low, C., Yip, Y., & Wu, T. H. (2006). Modelling and heuristics of FMS scheduling with multiple objectives. *Computers and Operations Research*, 33, 674-694.

Mastrolilli, M. & Gambardella, L. M. (2000). Effective neighbourhood functions for the flexible job shop problem. *Journal of Scheduling*, 3, 3-20.

Mati, Y. & Xie X. (2004). The complexity of two-job shop problems with multipurpose unrelated machines. *European Journal of Operational Research*, 152, 159-169.

Mesghouni, K., Hammadi, S., & Borne, P. (1998). On modeling genetic algorithms for flexible job-shop scheduling problems. *Studies in Information Control (Romania)*, 7, 37-47.

Moslehi, G. & Mahnam, M. (2011). A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics*, 129, 14-22.

Ong, Z. X., Tay, J. C., & Kwoh, C. K. (2005). Applying the clonal selection principle to find flexible job-shop schedules. In *Artificial Immune Systems* (pp. 442-455). Springer Berlin Heidelberg.

Park, B. J., Choi, H. R., & Kim, H. S. (2003). A hybrid genetic algorithm for the job shop scheduling problems. *Computers & industrial engineering*, 45, 597-613.

Roshanaei, V., Naderi, B., Jolai, F., & Khalili, M. (2009). A variable neighborhood search for job shop scheduling with set-up times to minimize makespan. *Future Generation Computer Systems*, 25, 654-661.

Roshanaei, V., Balagh, A. K. G., Esfahani, M. M. S., & Vahdani, B. (2010). A mixed-integer linear programming model along with an electromagnetism-like algorithm for scheduling job shop production system with sequence-dependent set-up times. *The International Journal of Advanced Manufacturing Technology*, 47, 783-793.

Saad, I., Hammadi, S., Benrejeb, M., & Borne, P. (2008). Choquet integral for criteria aggregation in the flexible job-shop scheduling problems. *Mathematics and Computers in Simulation*, 76, 447-462.

Stafford, E. F. & Tseng, F. T. (2006). The M×N Flowshop Problem with Separable, Sequence-independent Setup Times. *International Journal of Operations Research*, 3, 155-172.

Tanese, R. (1987). Parallel genetic algorithms for a hypercube. *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, Hillsdale, 177-183.

Tanese, R. (1989). Distributed genetic algorithms. *Proceedings of the 3rd International Conference on Genetic Algorithms*, San Francisco, 434-439.

Tavakkoli-Moghaddam, R., Azarkish, M., & Sadeghnejad-Barkousaraie, A. (2011). Solving a multi-objective job shop scheduling problem with sequence-dependent setup times by a Pareto archive PSO combined with genetic operators and VNS. *The International Journal of Advanced Manufacturing Technology*, 53, 733-750.

Tay, J. C., & Ho, N. B. (2008). Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Computers & Industrial Engineering*, 54, 453-473.

Wang, L., Zhou, G., Xu, Y., & Liu, M. (2012). An enhanced Pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 60, 1111-1123.

Wang, X., Gao, L., Zhang, C., & Shao, X. (2010). A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 51, 757-767.

Xing, L. N., Chen, Y. W., Yang, K. W. (2008). Double layer ACO algorithm for the multi-objective FJSSP. *New Generation Computing*, 26, 313-327.

Xing, L. N., Chen, Y. W., & Yang, K. W. (2009). Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling. *Applied Soft Computing*, 9, 362-376.

Zhang, G., Shao, X., Li, P., & Gao, L. (2009). An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. *Computers & Industrial Engineering*, 56, 1309-1318.

(Received December 11, 2014; accepted January 14, 2015)

Abstract

The flexible job-shop scheduling problem has received considerable attention in the production scheduling field. Setup operators are inseparable from the system of a small-volume and widely diverse producing. The work load balance among setup operators is important to increase the satisfaction at work. In this paper, a diversity-oriented multi-objective island based genetic algorithm is proposed to solve the FJSP considering the setup operators as a constraint. The total tardiness of each product and the difference of the setup number between setup operators are regarded as objectives and optimized using the Pareto-based method. When the proposed GA is applied to a test instance with moderate size of the problem complexity, it obtained the two exact solutions within a few seconds. Results show that the proposed algorithm, which was applied to the realistic production planning of the model enterprise, improves diversity of sub-solutions and the local search performance by the introduction of the diversity-oriented evaluation of each island, and emigration strategy among islands.