Flexible job shop scheduling using genetic algorithm and heuristic rules

Parinya KAWEGITBUNDIT* and Toru EGUCHI*
*Graduate School of Engineering, Hiroshima University
1-4-1 Kagamiyama, Higashi-Hiroshima 739-8527, Japan
E-mail: eguchi@hiroshima-u.ac.jp

Received 6 August 2015

Abstract
Job shop scheduling with the availability of more than one machine to perform an operation, also known as the flexible job shop scheduling problem, is computationally NP-hard. An efficient scheduling method is proposed here, using a genetic algorithm that incorporates heuristic rules. The scheduler’s goal is to minimize mean tardiness. There are two types of decision making required: job selection and machine selection. Combinations of five job selection and five machine selection heuristics are examined. Numerical experiments show that the combination of Yoda et al.’s (SL/RPN)+SPT rule for job selection and Eguchi et al.’s (WINQ+RPT+PT)×PT rule for machine selection provide the best performance under different shop conditions when incorporated into the genetic algorithm. It is also found that applying genetic algorithm only for either job selection or machine selection can generate good schedules, depending on conditions.

Key words: Job shop scheduling, Optimization, Genetic algorithms, Heuristics

1. Introduction

Job shop scheduling is a well-known combinatorial problem, and various optimization methods have been proposed in the literature. In many papers dealing with job shop scheduling, an operation can be processed only on a specific machine. However, in real-world situations there are generally multiple machines that can process a given operation, with differing processing times. Selecting which machine to perform an operation among those available is also an important consideration in trying to improve the performance of a scheduler. Job shops in which there are multiple machines that can process an operation are referred to as flexible or multipurpose-machine job shops. In general, such problems are intractable; even the task of scheduling two identical, parallel machines to minimize the maximum completion time is NP-hard (Lenstra et al., 1977). Flexible or multipurpose-machine job shop scheduling with more than three jobs and two machines is also NP-hard (Brucker et al., 1997). There are two types of flexible job shop (Chan et al., 2006): In the first, a job can have alternative operation sequences and multiple machines capable of each operation. In the second, jobs have a fixed operation sequence but there are still alternative machines for each operation. This paper deals with the latter type of problem. The performance measure adopted for scheduling is the mean job tardiness.

Many optimization approaches for flexible job shop scheduling have been proposed, such as exact solution algorithms, metaheuristic algorithms, and heuristic algorithms using priority rules. Exact algorithms such as branch-and-bound methods (Iwata et al., 1978) cannot optimally solve larger scale problems. Among the metaheuristic algorithms, many researcher have applied genetic algorithms (GAs) to scheduling with alternative machines (Candido et al., 1998; Jawahar et al., 1998; Norman and Bean, 1999; Morad et al., 1999; Lee and Kim, 2001; Moon et al., 2008; Pezzella et al., 2008; Shao et al., 2009; Li et al., 2010; Phanden et al., 2013). Although such algorithms can solve larger scale problems in reasonable time, the performance of the solutions deteriorates as the scale of problem grows. Heuristic algorithms using priority rules (Eguchi et al., 2006; Doh et al, 2013) have been developed and found to be effective in certain environments. Although they have a strong advantage in terms of calculation time, the quality of the solutions is not high in general.
In these situations, metaheuristic algorithms incorporating problem-specific knowledge have been proposed for searching out high-quality solutions in a reasonable amount of time. Eguchi et al. (2005) proposed a GA incorporating priority rules and showed that the method can generate better schedules than using a GA alone for larger scale problems. However, this approach considered only job selection and has not been applied to machine selection in flexible job shop scheduling. When using this method, it is necessary to find effective rules for incorporation into the GA. For scheduling with the mean tardiness of jobs as a performance metric, effective combinations of heuristic rules for both job and machine selection have to be specified under various scheduling conditions.

This paper proposes an efficient approach to flexible job shop scheduling using a GA that incorporates heuristic rules. For the purpose, various combinations of job and machine selection rules are evaluated under different scheduling conditions, and the best rule combination is determined. In flexible job shop scheduling, the relative importance of job and machine selection varies depending on scheduling conditions. In the proposed method, the GA can be applied only to one of the two selections, and the other selection is carried out using a heuristic rule alone. It is further shown that applying genetic algorithm only for either job selection or machine selection can generate good schedules, depending on scheduling conditions.

The rest of the paper is organized as follows: Section 2 formalizes the problem. Section 3 gives the heuristic rules under consideration. Section 4 explains the configuration of the GA, which is based on random key coding, and the incorporation of the heuristics. Section 5 describes numerical experiments conducted to examine the effectiveness of the proposed method. Finally, Section 6 summarizes the conclusions.

2. Job shop scheduling with alternative machines

We consider two types of job shop with alternative machines, as shown in Fig. 1. The first shop consists of \( N \) work centers. Each center \( W_q \) \((q \in \{1, 2, \ldots, N\})\) has \( L_q \) machines. The arrival time and the due date of a job \( J_i \) \((i = 1, 2, \ldots, I)\) is \( r_i \) and \( d_i \), respectively. Each job \( J_i \) requires a set of \( n_i \) operations \( o_{ij} \) \((j = 1, 2, \ldots, n_i)\), which are processed in order of increasing operation number \( j \). Operation \( o_{ij} \) is processed by the center \( W_{Sy} \) \((y \in \{1, 2, \ldots, N\})\). The routing of the operations through the work centers is flexible, but we assume that \( W_{Sy} \neq W_{S_{ij-1}} \) for \( j = 1, 2, \ldots, n_i - 1 \), namely, that no two successive operations of a job are processed in the same center. An operation to be processed in center \( W_q \) is performed on one of its \( L_q \) machines. The machine that processes the operation \( o_{ij} \) is denoted \( M_{ijk} \) \((k \in \{1, 2, \ldots, L_q\})\), and the processing time on the machine is \( p_{ijk} \).

We also consider a similar case, a random job shop with alternative machines. In this case, the shop consists of \( M \) machines. Each job \( J_i \) \((i = 1, 2, \ldots, I)\) requires \( n_i \) operations \( \{o_{ij} \,(j = 1, 2, \ldots, n_i)\} \), which again are processed in order of increasing \( j \). Each operation can be processed on one of \( R \) machines among the \( M \) machines in the shop. The specific combination of the \( R \) machines able to process an operation varies depending on the nature of the operation. The machine that processes operation \( o_{ij} \) is denoted \( M_{ijk} \) \((k \in \{1, 2, \ldots, R\})\), and the processing time is \( p_{ijk} \).

There are \( N \) work centers in the shop. Each center has \( L_q \) machines, and each operation can be processed on one machine in a center.

There are \( M \) machines in the shop. Each operation can be processed on one of \( R \) of these machines.

Fig. 1 Job shop configurations.
In both scenarios, each machine can process only one job at a time, and each job can be processed only on one machine at a time. Operations cannot be interrupted (non-preemption). Transportation times are not considered. The objective is to minimize the mean job tardiness $MT$, which is defined as follows:

$$MT = \frac{1}{I} \sum_{i=1}^{I} \max\left(0, C_i - d_i\right),$$  

where $C_i$ is the time required to complete job $J_i$.

3. Heuristic rules for job and machine selection

This section describes the procedure for generating schedules and the heuristic rules used in the procedure. We assume that each machine is equipped with an input buffer in which jobs wait to be processed. When a new job arrives on the shop floor, a machine is selected to process the first operation, and the job is transferred to its input buffer. When the job operation is finished, the machine to process the next operation is selected and the job is transferred to the input buffer of the selected machine. When a machine has become idle and there are jobs waiting to be processed in its input buffer, one of the jobs is selected for immediate processing. If there is only one waiting job, it is automatically started. This paper considers non-delay scheduling.

There are two types of decision to be made in this approach to scheduling: job selection and machine selection. When a machine has become idle and it has jobs waiting to be processed, one has to be selected for the next round of processing. When a new job arrives on the shop floor or a machine has finished processing a job operation, one of the machines that can process the next operation has to be selected. Heuristic rules are applied to these choices, and the overall scheduling performance depends on their effectiveness.

3.1 Rules for job selection

Various priority or dispatching rules have been proposed in the literature for the task of job selection. Because the objective here is to minimize mean job tardiness, five rules have been chosen for further examination. In each case, a priority index (PI) is calculated, and the job with the highest value is selected to be processed next.

3.1.1 ATC

The ATC rule (Vepsalainen and Morton, 1987) assigns a priority index $PI_{ijk}$ to each waiting job operation $o_{ij}$ as follows:

$$PI_{ijk} = \frac{1}{p_{ij}} \exp\left(-\frac{\max\{d_i - t - rpt_i - b(rpt_i - p_{ijk}), 0\}}{\kappa \bar{p}}\right),$$  

where $t$ is the current time, $\bar{p}$ is the average processing time of operations of jobs waiting to be processed on a particular machine, and $rpt_i$ denotes the time remaining for the processing of the job. In this paper, when calculating $rpt_i$, the processing time of the succeeding operation is taken to be the average of the processing times on the candidate machines for the operation. Here $\kappa$ and $b$ are adjustable parameters.

3.1.2 CR+SPT

The CR+SPT rule (Anderson and Nyirenda, 1990) calculates $PI_{id}$ for each waiting operation $o_{id}$ as

$$PI_{id} = \frac{1}{p_{id}} \left(\max\left[\frac{d_i - t}{rpt_i}, 1\right]\right)^{-1}.$$  

(3)
3.1.3 (SL/RPN)+SPT

The (SL/RPN)+SPT rule (Yoda et al., 2014) assigns a priority value for each waiting operation \( o_{ij} \) using

\[
PI_{ij} = \frac{1}{p_{ij}} \left( \max \left( \frac{(d_i - t - rpt_i)}{rpn_i} \right) + 1 \right)^{-1},
\]  

(4)

where \( rpn_i \) is the number of operations remaining to be completed for job \( J_i \).

3.1.4 SLACK

The SLACK rule assigns a priority value for each waiting job operation \( o_{ij} \) as

\[
PI_{ij} = \left[ \exp \left( \frac{d_i - t - rpt_i}{k} \right) \right]^{-1},
\]  

(5)

where \( k \) is a constant. When used alone, this rule simply selects the job with the smallest value of \( (d_i - t - rpt_i) \). The exponential function is introduced for incorporation of this rule into the GA dealt with in this paper.

3.1.5 EDD

The EDD rule’s priority index for each waiting operation \( o_{ij} \) is

\[
PI_{ij} = 1/d_i.
\]  

(6)

This rule selects the job with the earliest due date.

3.2 Rules for machine selection

In contrast to the rules for job selection, not many heuristics for machine selection have been developed in the literature. Here we examine five candidate rules for machine selection. Priority indexes are again assigned, and the machine with the highest priority value is selected.

3.2.1 PT

The PT rule selects the machine with the lowest processing time among the candidates. When a machine to process the next operation \( o_{ij} \) of a job has to be selected, the priority index is calculated for each candidate \( M_{ijk} \) using

\[
PI_{ijk} = 1/p_{ijk}.
\]  

(7)

3.2.2 NINQ

The NINQ rule selects the machine with the fewest jobs in its input buffer. When a machine is needed to process the next operation \( o_{ij} \) of a job, the priority index for each candidate \( M_{ijk} \) is calculated as

\[
PI_{ijk} = 1/(n_k + 1),
\]  

(8)

where \( n_k \) represents the number of jobs waiting in the input buffer of a machine \( M_{ijk} \) that can process \( o_{ij} \).
3.2.3 WINQ

The WINQ rule selects the machine with the smallest total processing time for all of the jobs waiting in its input buffer. When a machine to process the next operation of a job has to be selected, the index value for each candidate machine \( M_{ijk} \) is calculated as

\[
P_{ijk} = \frac{1}{\text{winq}_k},
\]

where \( \text{winq}_k \) represents the total of the imminent processing times of all jobs waiting in the input buffer of a machine \( M_{ijk} \) that can process the next operation.

3.2.4 WINQ+RPT+PT

The WINQ+RPT+PT rule selects the machine with the smallest total value of (1) processing times for all waiting jobs in its input buffer, (2) the remaining processing time \( \text{rptc}_k \) of the operation currently being processed, and (3) the processing time of the next operation if the operation is processed on the machine. The priority index is given by

\[
P_{ijk} = \frac{1}{(\text{winq}_k + \text{rptc}_k + p_{ijk})}.
\]

3.2.5 (WINQ+RPT+PT)×PT

The (WINQ+RPT+PT)×PT rule (Eguchi et al., 2006) combines the WINQ+RPT+PT and PT rules. This rule selects the machine with the smallest value of the times \( (1) + (2) + (3) \times (3) \) described in Section 3.2.4. The priority index is calculated as

\[
P_{ijk} = \frac{1}{(\text{winq}_k + \text{rptc}_k + p_{ijk})p_{ijk}}.
\]

The PT rule is effective at reducing total workload by selecting machines that can process operations with smaller processing times. NINQ and WINQ are the basic rules for workload balancing. The WINQ+RPT+PT rule is effective for more detailed workload balancing. The combined (WINQ+RPT+PT)×PT rule aims to both balance workload and reduce total workload (Eguchi et al., 2006).

4. Scheduling using a genetic algorithm

Our goal is to find an efficient scheduling method for job shops with alternative machines using a GA that incorporates heuristic rules. We next describe the GA used in this paper.

4.1 Random key based GAs

A variety of GAs have been proposed in the optimization literature. Among these, the algorithm proposed by Bean (1994), which uses random keys, is convenient for incorporating the heuristic rules described Section 3. The random keys can be treated as priority values for job selection. A gene \( j_{\text{gene}} \) is assigned to each operation \( o_j \), where \( j_{\text{gene}} \) takes real values between 0 and 1. When a job has to be selected for processing, the operation with the highest value of \( j_{\text{gene}} \) is chosen.

For machine selection, Norman and Bean (1999) proposed an approach based on machine number coding. Genes numbered 1, ..., \( k \) are used for machine selection, and their values directly correspond to the machines available to process the next operation. However, here we also adopt real numbers for coding the machine selection because of the convenience of incorporating heuristic rules. A gene \( m_{\text{gene}} \) is assigned to each candidate machine for an operation \( o_j \).
The value of \( m_{\text{gene}}_{ijk} \) takes real values between 0 and 1. When a machine has to be selected to process a job’s next operation, the machine with the highest value of \( m_{\text{gene}}_{ijk} \) is chosen.

A chromosome consists of the genes \( j_{\text{gene}}_{ij} \) and \( m_{\text{gene}}_{ijk} \) for all job operations. The generational transition and genetic operators are designed following Eguchi et al. (2005). The genes in the chromosomes or individuals in the first generation are generated randomly using a uniform distribution on \([0, 1]\). The fitness of an individual is calculated by carrying out a scheduling simulation using the genes as priority values for job and machine selection. An individual with lower mean tardiness is defined as having higher fitness. Figure 2 provides an overview of the generational transition. The best \( r\% \) individuals in the current population are copied to the next generation, that is, an elitist strategy is adopted. The other \((100 - r)\% \) individuals in the next generation are newly generated by applying a crossover operation in which two individuals are selected randomly from the current population. Then a random real number is generated using a uniform distribution on \([0, 1]\). If this is greater than \( p_c \), the gene for the operation is taken from an individual and copied to the offspring. Otherwise, the gene is taken from the other individual. Mutation is applied by changing each gene randomly with probability \( p_m \% \). However, mutation is not applied to the genes of the fittest individual. Therefore, the mean tardiness of the fittest individual either decreases or remains the same with generational transitions.

![Fig. 2 Generational transition in the GA.](image)

### 4.2 Incorporation of heuristic rules

The performance of the GA can be improved by incorporating problem-specific knowledge. Embedding heuristic rules found to be effective for scheduling into the GA can facilitate the search for optimal schedules, and these can be incorporated when evaluating the fitness of an individual. When selecting an operation to be processed next, the operation with the highest priority value is selected. If the rules are used alone, the priority values are calculated using their heuristic targets. If the GA is used alone as described in Section 4.1, the priority values are given by genes. With a heuristic rule for job selection incorporated into the GA, the priority values are given by the combination of genes and a job selection rule; the priority value \( j_{\text{priority}}_{ijk} \) for each waiting job operation \( o_{ij} \) is calculated following Eguchi et al. (2005):

\[
 j_{\text{priority}}_{ijk} = j_{\text{gene}}_{ij} \times \Pi_{ijk}. \tag{12}
\]

A heuristic rule for machine selection can be similarly incorporated into the GA; the priority value for each candidate machine \( M_{ijk} \) is calculated as follows:

\[
 m_{\text{priority}}_{ijk} = m_{\text{gene}}_{ijk} \times \Pi_{ijk}. \tag{13}
\]

When incorporating heuristic rules into the GA, an individual in which all the genes have the same value is included in the initial population. This value is set to 0.5 in this paper. The scheduling performance corresponding to this individual
is the same as that generated using the heuristic rules alone, because the large/small relation of priority values of Eqs. (12) and (13) are determined only by $P_{ijk}$. The best individual is copied to the next generation, and the mutation step is not applied to it. Therefore, the mean tardiness of a schedule obtained using this method is guaranteed to be smaller than or at least equal to that of a schedule generated using the heuristic rules alone.

5. Numerical experiments
5.1 Experimental conditions

Two types of scheduling problems are generated for performance evaluation: a job shop that consists of work centers and a random job shop. For the work-center shop, performance is examined under three different scheduling conditions: moderate load, a state of low machine utilization, and the imposition of tight job due dates. As a result, we examine scheduling performance using four conditions: the work-center shop with three conditions and the random job shop. Detailed conditions for each problem are described below.

In the work-center model, the shop floor consists of eight work centers ($N = 8$), and each center has two machines ($L = 2$). Any operation assigned to a work center can be processed on either of its machines. The total number of machines is thus 16. There are 100 jobs assigned. The number of operations $n_i$ for each job is set randomly to lie between 4 and 8. The order of work stations to be visited is determined randomly. The processing times for an operation on the two candidate machines are taken to be different. We assume that the processing time on the first machine is shorter than that on the second for all operations. The processing time on the first machine is sampled from a uniform distribution between 5 and 100. The processing time on the second machine is determined by multiplying the processing time on the first machine by a speed factor, which is uniformly distributed between 1.0 and 2.0.

The due date $d_i$ of job $J_i$ is set as follows:

$$d_i = r_i + k_i \sum_{j=1}^{n_i} \sum_{k=1}^{2} P_{ijk} \cdot \frac{2}{2}.$$  \hspace{1cm} (14)

This is based on the TWK method (Baker, 1984). The interval between the release time $r_i$ and the due date $d_i$ is set to be proportional to the total processing time of all the operations for the job. Here we estimate the processing time of an operation by averaging the processing times on the two candidate machines. The parameter $k_i$ is the due date tightness factor. For problems with moderate load, the tightness factor for each $J_i$ is taken from a uniform distribution between 1.5 and 3.0. Approximately 10%–15% of the jobs are tardy at this tightness when using the GA and heuristic rules. For problems with tight due dates, $k_i$ is set by using a uniform distribution between 1.0 and 2.0. Approximately 25%–30% of the jobs are tardy at this tightness.

Twice as many jobs as the number of machines in the shop are released to the floor at time zero. Additional jobs arrive on the shop floor at exponentially distributed random intervals. For conditions of moderate load, the arrival rate is set so that the average machine utilization reaches 80%–90%. For low machine utilization, the arrival rate is set so that the average machine utilization becomes about 60%–70%.

For random job shop problems, there are simply 16 machines and 100 jobs. The number of operations $n_i$ for each job is randomly chosen to lie between 4 and 8. Each operation can be processed on the first and the second (alternative) machine. The order of the first machine to process each operation is randomly determined under the condition that no successive operations are processed on the same machine. The processing time of an operation on the first machine is taken from the uniform distribution between 5 and 100. The second machine is selected randomly, and its processing time is chosen randomly between the range of the processing time on the first machine and twice the processing time on the first machine. The job due dates and arrival rates are set in the same way as for the work-center problems with moderate load; there are about 10%–15% tardy jobs, and the machine utilization is about 80%–90% when using the GA with heuristic rules.

Thirty scheduling problems are generated for each scheduling condition.
5.2 Results using heuristic rules alone

First, the scheduling performance using heuristic rules alone is examined for the four scheduling conditions. Figures 3–6 show the results. Twenty-five combinations using five job selection rules and five machine selection rules are tested. In this paper, the parameters for the heuristic rules are set by using preliminary experiments as follows: $\kappa = 1.5$ and $b = 1$ for the ATC rule, and $k_s = 100$ for SLACK. In each figure, the mean tardiness of jobs [Eq. (1)] when using each combination of job selection and machine selection rule is evaluated based on the mean value $\text{mean}_{\text{rules}}$ and the standard deviation $\text{diff.s.d.}_{\text{rules}}$ of the 30 problems as follows:

$$\text{mean}_{\text{rules}} = \frac{1}{30} \sum_{l=1}^{30} \text{MT}_{\text{rules},l},$$  \hspace{1cm} (15)

$$\text{diff.mean}_{\text{rules}} = \frac{1}{30} \sum_{l=1}^{30} (\text{MT}_{\text{rules},l} - \text{MT}_{\text{best rules},l}),$$  \hspace{1cm} (16)

$$\text{diff.s.d.}_{\text{rules}} = \sqrt{\frac{1}{30} \sum_{l=1}^{30} \left( \frac{(\text{MT}_{\text{rules},l} - \text{MT}_{\text{best rules},l}) - \text{diff.mean}_{\text{rules}}}{30} \right)^2}.$$  \hspace{1cm} (17)

In Eq. (15), $\text{MT}_{\text{rules},l}$ represents the mean tardiness of the $l$th problem using a combination of a job selection and a machine selection rule. $\text{MT}_{\text{best rules},l}$ represents the mean tardiness of the $l$th problem using the combination of job and machine selection rules with the smallest value of $\text{mean}_{\text{rules}}$ in each figure. Here $\text{diff.s.d.}_{\text{rules}}$ denotes the standard deviation of the difference between the general and the best combinations of rules. In the figures, the colored bars show the mean values, and the error bars show the standard deviations. The value on each bar represents $\text{mean}_{\text{rules}}$. The length of the error-bar lines corresponds to twice the value of $\text{diff.s.d.}_{\text{rules}}$.

Figures 3–5 show that the results using the PT rule for machine selection are very bad. This is natural, because the second machine in each work center is never used when using the PT rule. Figure 6, by contrast, shows that using the PT rule in a random job shop is not very bad. This is because the first machine for each job operation is randomly selected from all the machines in the shop.

![Fig. 3 Mean tardiness using heuristic rules alone for work-center problems under moderate load.](image-url)
Fig. 4  Same as Fig. 3, but for low machine utilization.

Fig. 5  Same as Fig. 3, but for tight due dates.

Fig. 6  Mean tardiness using heuristic rules alone for a random job shop.
Figures 3–6 indicate that the machine selection rules have a greater impact on performance than the job selection rules. For all conditions, the (WINQ+RPT+PT)×PT rule performs best. There are relatively small differences due to the job selection rules. Among them, the (SL/RPN)+SPT rule has the smallest mean tardiness for all the conditions. This rule can be considered as one of the best job selection rules if we have to select one rule. As a result, we conclude that the best combination of heuristic rules is (SL/RPN)+SPT plus (WINQ+RPT+PT)×PT rule when using heuristic rules alone.

5.3 Effectiveness of random key based coding for machine selection

In this paper, we adopt a GA using random key based coding not only for job selection but also for machine selection. To examine the effectiveness of this coding method for machine selection, the method is compared with the coding method based on machine numbers proposed by Norman and Bean (1999). In this experiment, the parameters in the GA are set as follows: $r = 20$, $p_c = 0.7$, $p_m = 1$. These values were determined based on preliminary experiments. The number of individuals and generations are 400 and 1000, respectively. The work-center problems with moderate load are solved using the two types of GAs described in Section 4.1. Figure 7 shows the generational transition of the GAs using the two coding methods for machine selection. Each curve in the figure is the average generational transition of the best individuals for 30 problems. This illustrates that the proposed random key based coding outperforms machine number based coding.

5.4 GA results incorporating heuristic rules

Next we examine the performance of the GA incorporating heuristic rules. Three types of GAs incorporating heuristic rules are compared: (1) the method in which the GA is applied both to job and machine selection, (2) the method in which the GA is applied only to job selection and machine section is carried out using an independent rule, and (3) the method in which the GA is applied only to machine selection and job selection is carried out using an independent rule. Each method is evaluated for each of the four problems types. The parameters of the GA are set as in the previous session.

5.4.1 Work-center job shops under moderate load

Figures 8–10 show the mean job tardiness obtained by using the GA incorporating heuristic rules for the work-center type problems with moderate load. For all three GA methods, the best results were obtained when using the (SL/RPN)+SPT rule for job selection and the (WINQ+RPT+PT)×PT rule for machine selection. For most combinations of heuristic rules, good performance was obtained when applying the GA only to machine selection in this condition. The overall best result was 7.7 in Fig. 10 when using the (SL/RPN)+SPT rule alone for job selection and applying the GA incorporating the (WINQ+RPT+PT)×PT rule to machine selection.

---

Fig. 7 Generational transition of the GAs using random key based coding and machine number based coding for machine selection
5.4.2 Work-center job shop with low machine utilization

Figures 11–13 show the results for work-center problems with low machine utilization. Results similar to those under moderate load were obtained, although the mean tardiness values are smaller than those of the moderate conditions. For all three methods of applying the GA, the best results were obtained when using the (SL/RPN)+SPT rule for job selection and (WINQ+RPT+PT)×PT for machine selection. For most combinations of the heuristic rules, good performance was also obtained when applying the GA only to machine selection in this condition. However, applying the GA only to job selection performs best when using the ATC rule or SLACK for job selection and the (WINQ+RPT+PT)×PT rule for machine selection. The overall best result was 6.2 in Fig. 13 when using the (SL/RPN)+SPT rule alone for job selection and applying the GA with the (WINQ+RPT+PT)×PT rule for machine selection.

5.4.3 Work-center job shop with tight due dates

Figures 14–16 show the results for work-center problems with tight due dates. Results similar to those under moderate load were obtained, although the mean tardiness values are larger than those of the moderate conditions. For all three methods of applying the GA, the best results were obtained when using the (SL/RPN)+SPT rule for job selection and (WINQ+RPT+PT)×PT for machine selection. For most combinations of the heuristic rules aside from using the (WINQ+RPT+PT)×PT rule for machine selection, good performance was also obtained by applying the GA only to machine selection in this condition. However, the overall best result was 22.7 in Fig. 15 when using the GA incorporating the (SL/RPN)+SPT rule for job selection and applying the (WINQ+RPT+PT)×PT rule alone to machine selection.

5.4.4 Random job shop

Figures 17–19 show the results for the random job shop. For all the three methods of applying the GA, the best results were obtained when using the (SL/RPN)+SPT rule for job selection and (WINQ+RPT+PT)×PT for machine selection. For most combinations of the heuristic rules, good performance was obtained when applying the GA only to machine selection in this condition. The overall best result was 3.4 in Fig. 19 when using the (SL/RPN)+SPT rule alone for job selection and applying the GA incorporating the (WINQ+RPT+PT)×PT rule for machine selection.
### Table

| Method                  | Mean Tardiness |
|-------------------------|----------------|
| GA((WINQ+RPT+PT) x PT) | 13.0, 17.4, 23.0 |
| GA(WINQ+RPT+PT)        | 15.3, 19.7, 25.0 |
| GA(WINQ)               | 17.4, 21.1, 26.5 |
| GA(NINQ)               | 19.5, 23.2, 28.8 |
| GA(PT)                 | 21.6, 25.2, 30.8 |

### Fig. 8
Mean tardiness when the GA is applied to both job and machine selection in work-center job shops under moderate load.

### Fig. 9
Mean tardiness when the GA is applied only to job selection in work-center job shops under moderate load.

### Fig. 10
Mean tardiness when the GA is applied only to machine selection in work-center job shops under moderate load.
Fig. 11  Mean tardiness when the GA is applied to both job and machine selection in work-center job shops with low machine utilization.

Fig. 12  Mean tardiness when the GA is applied only to job selection in work-center job shops with low machine utilization.

Fig. 13  Mean tardiness when the GA is applied only to machine selection in work-center job shops with low machine utilization.
Fig. 14  Mean tardiness when the GA is applied to both job and machine selection in work-center job shops with tight due dates.

Fig. 15  Mean tardiness when the GA is applied only to job selection in work-center job shops with tight due dates.

Fig. 16  Mean tardiness when the GA is applied only to machine selection in work-center job shops with tight due dates.
Kaweegitbundit and Eguchi, Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol.10, No.1 (2016)

Mean tardiness when the GA is applied to both job and machine selection in the random job shop problem.

Mean tardiness when the GA is applied only to job selection in the random job shop problem.

Mean tardiness when the GA is applied only to machine selection in the random job shop problem.
5.4.5 Results for smaller scale problems and unbalanced workloads

The results in Sections 5.4.1–5.4.4 show that effective application of the GA to the two selections depends on the details of the scheduling problem and the heuristic rules used. In most cases, applying the GA only to job or machine selection outperforms applying it to both. To examine the reason for this, two types of new problems are generated: small-scale problems and imbalanced workload problems.

We generated the small-scale problems by decreasing the size of the work-center problem with moderate load by setting the number of work centers to \( N = 2 \) and the randomly distributing the number of operations \( n_i \) for each job between 2 and 4. The other parameters are for the problems described in Section 5.4.1. Thirty problems were randomly generated. Figure 20 shows the generational transitions of the best individuals using the three GA types. The results show the mean values of the 30 problems. In this experiment, the (SL/RPN)+SPT rule is used for job selection and the (WINQ+RPT+PT)×PT rule is used for machine selection. Figure 20 indicates that applying the GA to machine selection is more effective than applying it to job selection in this condition. Applying the GA both to job selection and machine selection performed best in this case. This result suggests the reason why applying the GA both to job selection and machine selection did not perform best for the large-scale problems. When the scale of problem increases, the number of decisions for job selection and machine selection increases. Therefore, the solution space becomes large. It is difficult to search for an optimal schedule in the large solution space within a limited amount of time. Fixing one of the selections by using an effective heuristic rule alone can restrict the search space and can lead to better schedules as a result.

As a case in which the impact of searching job selections becomes relatively large, we generated problems with imbalanced workloads by modifying the work-center problem with moderate load. The processing times in two work centers among eight are determined using a uniform distribution between 5 and 200. Those two work centers become bottlenecks in the shop. Other conditions such as the number of machines are set as the same as those for the work center type jobs shop with moderate condition. Thirty problems are generated randomly. Figure 21 shows the generational transition of the best individuals using the three GA types. The results are shown by the mean values of the 30 problems. The (SL/RPN)+SPT rule is used for job selection and the (WINQ+RPT+PT)×PT rule is used for machine selection. Figure 21 shows that applying the GA only to job selection outperforms applying the GA only to machine selection in contrast to the results in Figs. 8–10. In this condition, the lengths of waiting queues in the buffers of certain machines become long and the job selection on the machines becomes more dominant on scheduling performance. In other words, the importance of job selection is relatively low when workload is balanced at the shop floor (Shimoyashiro et al., 1984). In this case, applying the GA both to job selection and machine selection performed best. However, the best method can vary depending on the balance of the impact of job selection and machine selection in the problem to be solved.
6. Conclusion

This paper has considered job shop scheduling with the availability of multiple machines to complete an operation. An efficient scheduling method using a genetic algorithm that incorporate heuristic rules has been proposed. The GA uses a random key coding approach not only for job selection but also for machine selection, whose effectiveness has been confirmed through numerical experiment. With this coding approach, heuristic rules can be easily incorporated into the GA. Five job selection rules and five machine selection rules were examined as heuristics for incorporation into the GA. Numerical experiments show that the combination of the (SL/RPN)+SPT rule for job selection and the (WINQ+RPT+PT)×PT rule for machine selection performs best for minimizing the mean tardiness of jobs in various conditions. Numerical experiments also show that applying the GA only to job selection or machine selection performs better than applying the GA both to job and machine selection. These experiments revealed that one of the reasons for this phenomenon is the complexity of searching job and machine selections simultaneously when the scale of the problem is large. The other reason is that the importance of job selection and machine selection varies depending on conditions. In general, if there are sufficient alternative machines available, the effect of job selection diminishes. Therefore, better schedules can be obtained by not applying the GA to job selection in most conditions. However, applying the GA to job selection is also effective under conditions in which job selection is important, such as when bottleneck machines exist.

References

Anderson, E. J. and Nyirenda, J. C., Two new rules to minimize tardiness in a job shop, International Journal of Production Research, Vol.28, No.12 (1990), pp.2277–2292.
Baker, K.R., Sequencing rules and due-date assignments in a job shop, Management Science, Vol.30, No.9 (1984), pp.1093–1104.
Bean, J. C., Genetic algorithms and random keys for sequencing and optimization, ORSA Journal on Computing, Vol.6, No.2 (1994), pp.154–160.
Brucker, P., Jurisch, B. and Krämer, A., Complexity of scheduling problems with multi-purpose machines, Annals of Operations Research, Vol.70 (1997), pp.57–73.
Candido, M. A. B., Khator, S. K. and Bracia, R. M., A genetic algorithm based procedure for more realistic job shop scheduling problems, International Journal of Production Research, Vol.36, No.12 (1998), pp.3437–3457.
Chan, F.T.S., Wong, T.C. and Chan, L.Y., Flexible job-shop scheduling problem under resource constraints, International Journal of Production Research, Vol.44, No.11 (2006), pp.2071–2089.
Doh, H.H, Yu, J.M, Kim, J.S., Lee, D.H. and Nam S.H., A priority scheduling approach for flexible job shops with multiple process plans, International Journal of Production Research, Vol.51, No.12 (2013), pp.3748–3764.
Eguchi, T., Oba, F. and Kozaki, S., Dynamic Scheduling Using the Mixture of a Genetic Algorithm and a Priority Rule, Transactions of the Japan Society of Mechanical Engineers Series C, Vol.71, No.703 (2005), pp.1047-1053. (in Japanese)

Eguchi, T., Oba, F., Toyooka, S. and Sato, Y., Machine selection rule for dynamic job shop having alternative machines with different processing times, Journal of the Japan Society for Precision Engineering, Vol.72, No.4 (2006), pp.459–464. (in Japanese)

Iwata, L., Murotsu, Y., Oba, F., Uemura, T. and Okushima, K., Optimization of selection of machine-tools, loading sequence of parts and machining conditions in job-shop type machine systems, Annals of the CIPR, Vol.27, No.1 (1978), 447–451.

Jawahar, N., Aravindan, P. and Ponnambalam, A genetic algorithm for scheduling flexible manufacturing systems, International Journal of Advanced Manufacturing Technology, Vol.14 (1998), pp.588–607.

Lee, H. and Kim, S.S., Integration of process planning and scheduling using simulation based genetic algorithm, International Journal of Advanced Manufacturing Technology, Vol.18 (2001), pp.586–590.

Lenstra, J.K., Rinnooy Kan, A.H.G and Brucker, P., Complexity of machine scheduling problems, Annals of Discrete Mathematics, Vol.1 (1977), pp.343–362.

Li, X., Shao, X., Gao, L. and Qian, W., An effective hybrid algorithm for integrated process planning and scheduling, International Journal of production economics, Vol.126 (2010), pp.289–298.

Moon, I., Lee, S. and Bae, H., Genetic algorithm for job shop scheduling problems with alternative routing, International Journal of Production Research, Vol.46, No.10 (2008), pp.2695–2705.

Morad, M. and Zalzala, A., Genetic algorithm in integrated process planning and scheduling, Journal of Intelligent Manufacturing, Vol.10 (1999), pp.169–179.

Norman, B. A. and Bean, J. C., A Genetic algorithms methodology for complex scheduling problems, Naval Research Logistics, Vol.46 (1999), pp.199–211.

Pezzella, F., Morganti, G. and Ciaschetti, G., A genetic algorithm for the flexible job-shop scheduling problem, Computers & Operations Research, Vol.35 (2008), pp.3202–3212.

Phandan, R.K., Jain, A. and Verma, R., An approach for integration of process planning and scheduling, International Journal of Computer Integrated Manufacturing, Vol.26, No.4 (2013), pp.284–302.

Shao, X., Li, X., Gao, L. and Zhang, C., Integration of process planning and scheduling–A modified genetic algorithm-based approach, Computers & Operations Research, Vol.36 (2009), pp.2082–2096.

Shimoyasiro, S., Isoda, K. and Awane, H., Input scheduling and load balance control for a job shop, International Journal of Production Research, Vol.22, No.4 (1984), pp.597–605.

Vepsalainen, A. P. J. and Morton, T. E., Priority rules for job shops with weighted tardiness costs, Management science, Vol.33, No.8 (1987), pp.1035–1047.

Yoda, M., Eguchi, T. and Murayama, T., Job shop scheduling for meeting due dates and minimizing overtime using genetic algorithm incorporating new priority rules, Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol.8, No.5 (2014), Paper No.14–00076.