Journal of the Operations Research
Society of Japan
Vol. 41, No. 2, June 1998

INTERACTIVE DECISION SYSTEM IN STOCHASTIC MULTIOBJECTIVE SCHEDULING TO MINIMIZE THE EXPECTED VALUE AND VARIANCE OF TOTAL FLOW TIME

Hiroyuki Nagasawa Chue Shing
Osaka Prefecture University

(Received September 20, 1996; Final February 2, 1998)

Abstract Stochastic two-objective scheduling to minimize both the expected value and the variance of total flow time is one of the most interesting subjects for shop-floor managers. This paper develops a heuristic method for generating a set of nondominated schedules to this two-objective scheduling problem in a parallel machine case, on the basis of the heuristic method proposed for a single machine case by Jung *et al.*(1990). Introducing some new categories of the set of nondominated schedules, this paper proposes an interactive system (ISMSS) for selecting a preferred schedule from among the set of nondominated schedules.

1. Introduction

In actual scheduling, shop floor managers often face the difficult task of evaluating which schedule is the best under multiple criteria. The number of papers on multiobjective scheduling remains still little compared with the case of conventional single-objective scheduling, although a long time has passed since Smith[19] initiated dual criteria scheduling in 1956. Van Wassenhov and Gelders[20] developed an efficient algorithm to generate a set of nondominated schedules to the two-objective single-machine scheduling problem to minimize both the mean flow time and the maximum tardiness, on the basis of Smith's algorithm. Nelson et al.[14] summarized multiobjective single-machine scheduling research and developed some tree procedures for generating a set of nondominated schedules. Huckert et al.[6] formulated a mathematical model of six-objective, M-machine job shop scheduling and proposed an interactive method to find a compromise schedule. Sen et al.[18], Daniel et al.[3] and Rajendran[16] proposed various time-consuming branch-and-bound algorithms to some multiobjective flowshop scheduling problems.

Various efficient heuristics for generating a set of nondominated schedules have been also developed recently. For example, Rajendran[17] developed some heuristic algorithms to multiobjective flowshop scheduling problems. Morizawa et al.[12] developed a generalized efficient method for generating a set of nondominated schedules and for searching a preferred schedule in multiobjective flowshop scheduling. Murata et al.[13] applied a genetic algorithm(GA) to some multiobjective flowshop scheduling problems.

On the other hand, multiobjective scheduling area has been also widened to some new field such as fuzzy-related scheduling and stochastic scheduling. For example, Ishii et al.[7] formulated their unique two-objective, single-machine scheduling problem to minimize the maximum lateness and to maximize a satisfaction level with respect to fuzzy precedence relation between any pair of jobs and proposed a method for generating a set of nondominated schedules. Jung et al.[9] provided an efficient algorithm based on a pairwise-job-interchange method to a stochastic two-objective scheduling problem to minimize both the expected value and variance of total flow time. Forst[4] dealt with bicriterion stochastic scheduling to minimize the sum of the expected total weighted tardiness and the expected total weighted flow time.

The major problem which should be resolved in multiobjective scheduling is how to select a preferred schedule from among a set of nondominated schedules without a lot of efforts for a decision maker to judge which schedule is better than others.

In solving multiple criteria decision making problems, Korhonen et al.[11] distinguished the following three solution principles: (i) Prior articulation of preferences, (ii) Interactive articulation of preferences and (iii) Posterior articulation of preferences. In multiobjective scheduling area, most papers are related to the third principle, but few methods have been proposed for selecting a preferred schedule from among the set of nondominated schedules. Since the set of nondominated schedules sometimes includes some hundreds of nondominated schedules, it is hard for a decision maker to select a preferred schedule from among such a lot of nondominated schedules by comparing the criterion vector values of nondominated schedules with each other.

Fortunately, in stochastic multiobjective scheduling, we can use some stochastic meaning of each criterion vector value for selecting a preferred schedule as additional information. Therefore, we present a new decision support system, called " Interactive Stochastic Multiobjective Scheduling System(ISMSS)," in stochastic multiobjective scheduling to minimize both the expected value and variance of total flow time initiated by Jung et al.[9].

Stochastic variation of processing times affects the completion times of jobs, and then the operation schedule in the succeeding stages. If the variance of total flow time is small, the difference between the planned schedule and the actual schedule becomes small, lightening the burden of rescheduling task in the succeeding stages. Otherwise, the difference becomes large, resulting in a lot of tardiness or requiring frequent rescheduling in the succeeding stages to reduce the tardiness. However, to minimize the variance of total flow time itself does not guarantee to minimize the expected value of total flow time but sometimes follows a large expected value of total flow time. Simple pairwise comparison of criterion vector values of nondominated schedules does not always help a decision maker reduce the set of nondominated schedules. Instead, we assess each criterion vector value by using the concept of the upperward 100α percentile, and derive the dominance relationships between nondominated schedules under an allowable excess probability α. If the the upperward 100α percentile of a nondominated schedule is larger than that of the other nondominated schedule, the former schedule is judged "worse" than the latter schedule under a given probability α. Using this relationship, we can reject many "worse" nondominated schedules from the set of original nondominated schedules for the range of α specified by the decision maker. Furthermore, in selecting the perferred schedule, it is easier for the decision maker to assess a pair of α and the upperward 100α percentile than to assess a pair of the expected value and variance of total flow time directly.

Therefore, this paper presents some new concepts of nondominated schedules related to the upperward 100α percentile, and discusses the relationships between the original nondominated schedules and the newly derived nondominated schedules. Exploiting the relationships, we propose the ISMSS for the decision maker to select the preferred schedule from among the set of nondominated schedules. Through this ISMSS, we can show the decision maker a lot of original nondominated schedules on a computer display in an aggregated manner, and automatically present the relationships between these nondominated schedules under various values of α. Tasks necessary for the decision maker are only to specify the range of allowable excess probability α and finally to select the preferred schedule from among a few remaining candidate schedules. These tasks are much easier for the decision maker to do than those of repeating direct comparisons of the criterion vector values of nondominated schedules in a usual way, reducing the difficulty in judgements for the decision maker to select the preferred schedule.

For this purpose, we first describe "ISMSS" in a single-machine case. Secondly, we propose a heuristic algorithm to generate a set of nondominated schedules in a parallel-machine case, and extend the interactive system to the parallel machine case. Some numerical examples are shown to demonstrate the effectiveness of the proposed system.

2. Stochastic Multiobjective Scheduling

According to Abdelaziz et al.[1], the generalized form of the "Multiobjective Stochastic Linear Programming(MSLP)" problem is defined as follows:

$$\min\{\boldsymbol{C}(\omega)\boldsymbol{x} \mid \boldsymbol{A}(\omega)\boldsymbol{x} \leq \boldsymbol{b}(\omega), \omega \in \Omega\}, \tag{2.1}$$

where $\boldsymbol{C}(\omega)$ and $\boldsymbol{A}(\omega)$ are random matrices, and $\boldsymbol{b}(\omega)$ is a random vector defined on some probability space $(\Omega, \mathcal{B}, \mathcal{P})$. If the constraint can be replaced with deterministic one, and if the probability space Ω can be approximated by a small number of scenarios $s \in \mathcal{S} = \{1, 2, \cdots, S\}$, the following formulation is obtained:

$$\min\{\widetilde{\boldsymbol{C}}\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b},\ \mathrm{Prob}\{\widetilde{\boldsymbol{C}} = \boldsymbol{C}^s\} = p_s,\ s \in \mathcal{S}\}. \tag{2.2}$$

Joshi[8] formulated " Stochastic Programming Problem(SPP)" as

$$\min\{E[f(\boldsymbol{x}, \boldsymbol{C}(\omega))] \mid \boldsymbol{g}(\boldsymbol{x}) \geq \mathbf{o},\ \omega \in \Omega\}, \tag{2.3}$$

or introducing the scenarios, he also gave another expression of SPP as

$$\min\{E[f(\boldsymbol{x}, \widetilde{\boldsymbol{C}})] \mid \boldsymbol{g}(\boldsymbol{x}) \geq \mathbf{o},\ \mathrm{Prob}\{\widetilde{\boldsymbol{C}} = \boldsymbol{C}^s\} = p_s,\ s \in \mathcal{S}\}, \tag{2.4}$$

where $E(\cdot)$, $\boldsymbol{g}(\cdot)$ and $f(\cdot)$ denote the expected value, a vector function, and a scalar objective function, respectively.

Using these formulations, we can formulate a stochastic multiobjective scheduling problem as follows:

$$\min\{\boldsymbol{f}(\pi, \boldsymbol{C}(\omega)) \mid \pi \in \Pi,\ \omega \in \Omega\}, \tag{2.5}$$

where $\boldsymbol{f}(\cdot)$ denotes a scheduling criterion vector function and Π denotes a set of feasible schedules, π. The element of $\boldsymbol{C}(\omega)$, denoted by $c_{ij}(\omega)$, is a random processing time of job J_i at machine M_j in this case. Taking the expectation of $\boldsymbol{f}(\cdot)$, we can derive the following form:

$$\min\{\boldsymbol{E}[\boldsymbol{f}(\pi, \boldsymbol{C}(\omega))] \mid \pi \in \Pi,\ \omega \in \Omega\}, \tag{2.6}$$

where $\boldsymbol{E}[\boldsymbol{f}(\cdot)]$ denotes a expected value of the vector function $\boldsymbol{f}(\cdot)$ whose element is defined by $E[f_i(\cdot)]$. The chance constrained scheduling problem solved by Kise et al.[10] is a single-objective case of the stochastic scheduling problem defined by Eq.(2.5) where the objective is to maximize the number of early jobs being completed prior to its due date with a probability greater than or equal to α. Of course, we can introduce some stochastic constraints to our scheduling problem as follows:

$$\min\{\boldsymbol{f}(\pi, \boldsymbol{C}(\omega)) \mid \boldsymbol{g}(\boldsymbol{C}(\omega), \omega) \geq \mathbf{o},\ \pi \in \Pi, \omega \in \Omega\}, \tag{2.7}$$

or

$$\min\{\boldsymbol{E}[\boldsymbol{f}(\pi, \boldsymbol{C}(\omega))] \mid \boldsymbol{g}(\boldsymbol{C}(\omega), \omega) \geq \mathbf{o},\ \pi \in \Pi, \omega \in \Omega\}, \tag{2.8}$$

where $\boldsymbol{g}(\boldsymbol{C}(\omega), \omega)$ denotes a random vector function representing the stochastic constraints such as random machine failure, random due date and random arrival time. We call the scheduling problem defined by one of Eqs.(2.5) through (2.8) the "Stochastic Multiobjective Scheduling Problem(SMSP)."

In this paper, we consider the following stochastic multiobjective scheduling problem initiated by Jung et al.[9]:

$$\text{P1: } \min\left\{ \begin{pmatrix} E[F(\pi)] \\ V[F(\pi)] \end{pmatrix} \middle| \ \pi \in \Pi \right\}, \tag{2.9}$$

where $F(\pi)$, $E[F(\pi)]$ and $V[F(\pi)]$ denote a total flow time, the expected value and the variance of total flow time, respectively. Precisely speaking, we have to write $F(\pi, \omega)$ instead of $F(\pi)$ and to add the constraint of $\omega \in \Omega$ into Eq.(2.9). However, it is not necessary to give always such a precise expression if the scheduling problem is specified as a stochastic scheduling problem, and it should be rather omitted for simplicity. Eq.(2.9) is included in a form of Eq.(2.6), because $V[F(\pi)] = E[\{F(\pi) - E[F(\pi)]\}^2]$.

The set of nondominated schedules to P1 is denoted by $\mathcal{N}$. Jung et al.[9] showed that their heuristic algorithm based on a job-interchange method provided the approximate set of nondominated schedules very close to the true set of nondominated schedules derived through the branch-and-bound method. Therefore, we employ their method to generate the set of nondominated schedules to P1. Since "the smaller the better" principle holds in these criteria, the "preferred schedule" to be selected by a decision maker exists in the set of nondominated schedules $\mathcal{N}$. The difficulty is how to select the preferred schedule from among these nondominated schedules.

Unlike the usual deterministic multiobjective scheduling, we can use a stochastic meaning in selecting a preferred schedule from among the set of nondominated schedules in stochastic multiobjective scheduling. Therefore, we propose an interactive decision system to help a decision maker select a preferred schedule by showing the stochastic meaning of each criterion vector value in the above two-objective stochastic scheduling problem. First we present some new categories related to the set of nondominated schedules $\mathcal{N}$.

2.1. Upperward 100 α percentile minimum schedule

The schedule minimizing $E[F(\pi)]$ is derived only by focusing on the statistical " mean " of the total flow time and it allows the total flow time to exceed the expected value with a probability of 50%. However, some managers wish to decrease the excess probability down to a smaller value, for instance 10%. Let y be the upperward 100 α percentile, that is, the total flow time $F(\pi)$ exceeds y with a probability of α. If managers wish to minimize the time y, this decision problem is formulated as

$$\text{P2: } \min\{y \mid \Pr.\{\mathrm{F}(\pi) \leq y\} \geq 1 - \alpha,\ \pi \in \Pi\}. \tag{2.10}$$

From our experience of time measurements in a real factory[15], most of human processing times follow Erlang distributions with various location and shape parameters, and the rest follows a normal distribution. The coefficient of variation varies within tha range of 0.05 ~ 0.40 according to the property of each operation: 0.05 ~ 0.20 for easy tasks; 0.10 ~ 0.25 for normal tasks; 0.20 ~ 0.40 for difficult tasks. According to the central limit theorem, a distribution of the sum of independent variables converges into a normal distribution as the number of variables to be summed increases as long as the value of each variable does not give a large effect on its sum. We can apply this theorem to our case because the total flow time is the weighted sum of independent processing times and because each processing time gives a small amount of effect on the total flow time itself in usual cases of factories. Therefore, we assume that the total flow time follows a normal distribution with mean $E[F(\pi)]$ and variance $V[F(\pi)]$, no matter what kind of distribution each processing time follows.

Under this assumption, using the relationship of $\Phi(u_\alpha) = 1 - \alpha$ where $\Phi(\cdot)$ is a c.d.f. of a standard normal distribution and u_α is the upperward $100\alpha\%$ point in the standard normal distribution, we can replace y in Eq.(2.10) with $E[F(\pi)] + u_\alpha\sqrt{V[F(\pi)]}$, resulting in the following problem P3 equivalent to P2:

$$\text{P3: } \min\left\{E[F(\pi)] + u_\alpha\sqrt{V[F(\pi)]} \;\middle|\; \pi \in \Pi\right\}. \tag{2.11}$$

P3 is also equivalent to the problem to minimize the weighted sum of the original two objective functions in P1(since minimizing $V[F(\pi)]$ is equivalent to minimizing $\sqrt{V[F(\pi)]}$, we replace with $\sqrt{V[F(\pi)]}$ in P1 from this point on). The optimal schedule to this problem is called the " upperward 100 α percentile minimum schedule," denoted by $\pi(\alpha)$, and it can be obtained by searching the nondominated schedules in $\mathcal{N}$, because the weighted sum minimization problem defined by P3 gives an element in $\mathcal{N}$ if $u_\alpha > 0$. If we set $u_\alpha = 0$(corresponding to $\alpha = 0.5$) in P3, we get the simple problem, min $\{E[F(\pi)] \mid \pi \in \Pi\}$. Since the " upperward 50% point minimum schedule $\pi(0.5)$ " is a solution to P3 with $\alpha = 0.5$, $\pi(0.5)$ is also an " optimal schedule to minimize the expected value of total flow time $E[F(\pi)]$" (For simplicity, if $\pi(0.5)$ is not unique, we pick up the one with the minimum value of $V[F(\pi)]$ from such schedules as $\pi(0.5)$ so that $\pi(0.5) \in \mathcal{N}$). These two expressions denote the same schedule but the meaning is slightly different, that is, $\pi(0.5)$ is derived by focusing on the upperward 50% point. This is important when selecting a preferred schedule by considering the stochastic meaning of the criterion vector value.

2.2. Combined nondominated schedules

For a given value of u_α(or α), we can obtain $\pi(\alpha)$ as a solution to P3. However, this solution sometimes results in a large value of $V[F(\pi)]$, especially when u_α is close to zero(or α is close to 0.5). If we want to reduce the value of $V[F(\pi)]$, the upperward 100α percentile itself cannot always provide a good schedule with a small value of $V[F(\pi)]$. Therefore, we need to analyze the relationships between the minimization of the upperward 100α percentile of the total flow time and the minimization of the variance of the total flow time under a given u_α(or α). This analysis can be implemented by replacing $E[F(\pi)]$ with $E[F(\pi)] + u_\alpha\sqrt{V[F(\pi)]}$ in the original two objective scheduling problem P1, resulting in the following:

$$\text{P4: } \min\left\{\begin{pmatrix} E[F(\pi)] + u_\alpha\sqrt{V[F(\pi)]} \\ \sqrt{V[F(\pi)]} \end{pmatrix} \;\middle|\; \pi \in \Pi \right\}. \tag{2.12}$$

It is noticeable that P1 is a special case when $u_\alpha = 0$ (or $\alpha = 0.5$) in P4. The set of nondominated schedules to P4 is called the set of " combined nondominated schedules," denoted by $\mathcal{N}_c(\alpha)$. Obviously, $\mathcal{N}_c(\alpha) \subseteq \mathcal{N}$ and $\mathcal{N}_c(0.5) \equiv \mathcal{N}$. For any π_i in $\mathcal{N}$, if there is a schedule π_j in $\mathcal{N}$ satisfying both $V[F(\pi_j)] < V[F(\pi_i)]$ and$E[F(\pi_j)] + u_\alpha\sqrt{V[F(\pi_j)]} \leq E[F(\pi_i)] + u_\alpha\sqrt{V[F(\pi_i)]}$, then π_i is dominated by the π_j. Therefore, except for the nondominated schedule with the minimum variance, any schedule π_i in $\mathcal{N}$ becomes a dominated schedule when u_α is larger than or equal to $u_\alpha(\pi_i)$ defined by

$$u_\alpha(\pi_i) = \min\left\{ \frac{E[F(\pi_j)] - E[F(\pi_i)]}{\sqrt{V[F(\pi_i)]} - \sqrt{V[F(\pi_j)]}} \,\middle|\, V[F(\pi_i)] > V[F(\pi_j)],\ \pi_j \in \mathcal{N} \right\}. \quad (2.13)$$

Using this $u_\alpha(\pi_i)$, we obtain the set of nondominated schedules to P4 such as
$\mathcal{N}_c(\alpha) = \mathcal{N} \setminus \{\pi_i \mid u_\alpha(\pi_i) \leq u_\alpha,\ \pi_i \in \mathcal{N}\}$.

As α decreases(u_α increases), the number of elements in $\mathcal{N}_c(\alpha)$ decreases, and when $\alpha \to 0(u_\alpha \to \infty)$, only one element with the minimum variance remains in $\mathcal{N}_c(\alpha)$. The relationship between $\mathcal{N}_c(\alpha)$ and the set of $\pi(\alpha)$ is expressed by
$\{\pi(\alpha) \mid u_\alpha(\pi(\alpha)) > u_\alpha,\ \pi(\alpha) \in \mathcal{N}\} \subseteq \mathcal{N}_c(\alpha)$,
because $\mathcal{N}_c(\alpha)$ includes some schedules that can not be generated by the weighted sum minimization method(to obtain a set of solutions to P3 for various values of $u_\alpha > 0$).

2.3. Upperward 100 α% combined nondominated schedules

In some cases, managers wish to minimize both y and α, resulting in the following formulation:

$$\text{P5: } \min\left\{ \begin{pmatrix} y \\ \alpha \end{pmatrix} \,\middle|\, \text{Pr.}\{\text{F}(\pi) \leq y\} \geq 1 - \alpha,\ \pi \in \Pi \right\}. \quad (2.14)$$

Under the assumption that the total flow time follows a normal distribution, introducing $\Phi(u_\alpha) = 1 - \alpha$, we obtain another expression of P5 as

$$\text{P6: } \min\left\{ \begin{pmatrix} E[F(\pi)] + u_\alpha\sqrt{V[F(\pi)]} \\ u_\alpha \end{pmatrix} \,\middle|\, \pi \in \Pi \right\}. \quad (2.15)$$

The set of nondominated schedules to P6 is called the set of " upperward 100 α% combined nondominated schedules," denoted by $\mathcal{N}^\alpha$, which is also included in $\mathcal{N}$. All the elements of $\mathcal{N}^\alpha$ can be obtained by finding $\pi^\alpha(u)$ for any $u(\geq 0)$ such that
$V[F(\pi^\alpha(u))] = \max\{V[F(\pi)] \mid u_\alpha(\pi) > u, \pi \in \mathcal{N}\}$.

Therefore, we get the following relationships:
$\mathcal{N}^\alpha = \{\pi^\alpha(u) \mid 0 \leq u\}$, or equivalently, $\mathcal{N}^\alpha = \{\pi(\alpha) \mid 0 < \alpha \leq 0.5\}$.

3. Selection of the Preferred Schedule

Since a preferred schedule is included in the set of nondominated schedules, we define an initial set of candidates for a preferred schedule as the set of whole nondominated schedules $\mathcal{N}$. Basic idea for selecting a preferred schedule is to reduce the set of candidates considering the stochastic meaning of the criterion vector value of each nondominated schedule.

For an illustrative purpose, consider a 10-job, single-machine problem with random processing time whose mean μ_i and standard deviation σ_i are given by pattern 1 in Table 1. In this case, we obtained 13 nondominated schedules to the original problem P1 as shown in

Table 1. Input data of μ_i, σ_i and σ_i/μ_i of processing time of each job

Job	pattern 1			pattern 2		
i	μ_i	σ_i	σ_i/μ_i	μ_i	σ_i	σ_i/μ_i
1	150	17	0.11	150	15	0.10
2	130	23	0.18	130	20	0.15
3	140	15	0.11	140	17	0.12
4	150	20	0.13	150	23	0.15
5	180	38	0.21	180	14	0.08
6	190	36	0.19	190	10	0.05
7	170	25	0.15	170	19	0.11
8	120	14	0.12	120	38	0.32
9	100	10	0.10	100	25	0.25
10	110	19	0.17	110	36	0.33

Table 2. The set of nondominated schedules for the single-machine case (pattern 1: "*" denotes the element of $\mathcal{N}^{\alpha}$)

No.	$E[F(\pi)]$	$\sqrt{V[F(\pi)]}$	$u_\alpha(\pi)$
1*	7110	334.9	1.68
2*	7120	328.9	2.33
3*	7130	324.6	4.89
4	7140	323.9	2.93
5*	7150	320.5	9.38
6	7160	319.8	7.98
7	7170	318.7	7.24
8*	7180	317.3	10.9
9	7190	316.6	8.82
10	7200	315.5	14.2
11*	7210	314.8	26.8
12	7240	314.0	14.1
13	7250	313.3	∞

Table 2, using the heuristic algorithm proposed by Jung et al.[9]. The sets of nondominated schedules $\mathcal{N}$, $\mathcal{N}_c(\alpha)$ and $\mathcal{N}^{\alpha}$ are expressed in Figure 1. In the left-hand-side of Figure 1, the set of nondominated schedules to P1, $\mathcal{N}$, is plotted as a series of ◇ at the bottom of Figure 1. It is obvious that $\mathcal{N}_c(0.5) = \mathcal{N}$. The series of marks □, △ and ○ represent the sets $\mathcal{N}_c(\alpha)$ for $\alpha = 0.1$, 0.05 and 0.01, respectively. The upperward 100 α percentile minimum schedules $\pi(\alpha)$ for $\alpha = 0.5, 0.1, 0.05$, and 0.01 are positioned at the right edge of the corresponding set $\mathcal{N}_c(\alpha)$ in Figure 1. The set $\mathcal{N}^{\alpha}$ is composed of all the schedules $\pi(\alpha)$ for $0 < \alpha \leq 0.5$, which is also plotted in the right-hand-side of Figure 1, where “ No.1 ” denotes the nondominated schedule No.1 listed in Table 2. The No.4-schedule in Table 2 cannot become the upperward 100 α percentile minimum schedule for any α, because this schedule is dominated by No.5-schedule for $u_\alpha \geq 2.93$(the No.4-schedule is not included in $\mathcal{N}(\alpha)$ for $u_\alpha \geq 2.93$(or $\alpha \leq 0.017$) as shown in Figure 1: compare △ with ○) and is not optimal for P2 nor P3 when $u_\alpha < 2.93$.

Two frames corresponding to $\alpha = 0.05$ and 0.2 are drawn in Figure 1. The outer frame drawn by the dashed line in Figure 1 for $\alpha = 0.05$ shows that any schedule in

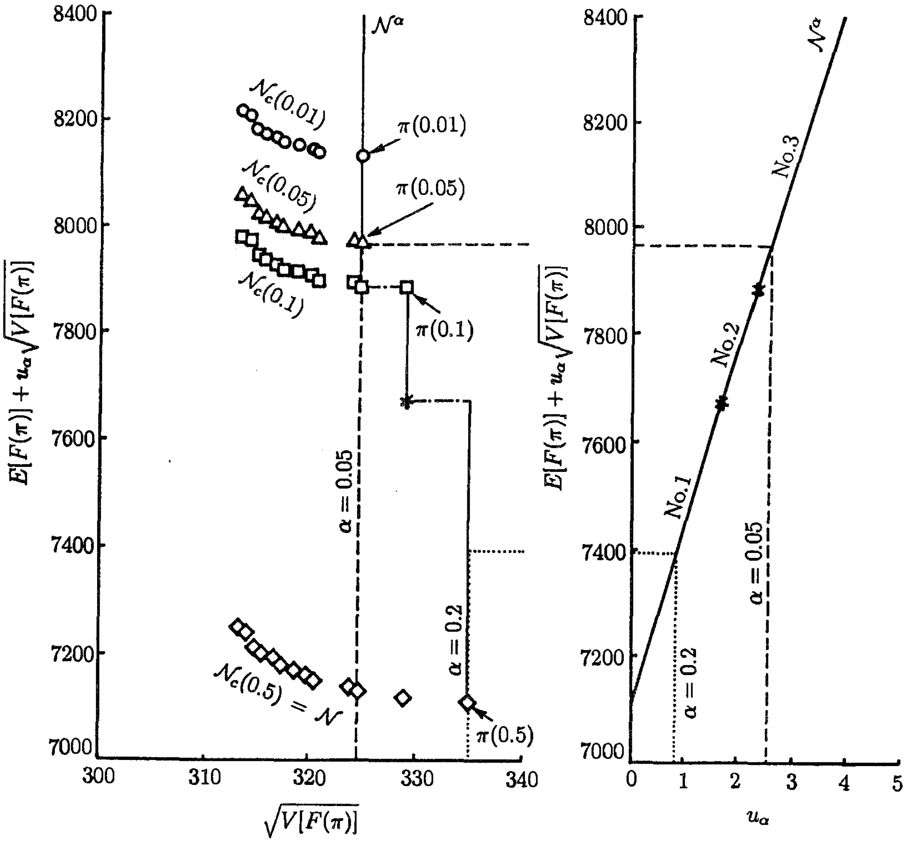


Figure 1. Relationships among the sets of nondominated schedules, $\mathcal{N}$, $\mathcal{N}_c(\alpha)$ and $\mathcal{N}^\alpha$

$\mathcal{N}_c(0.05) \setminus \{\pi(0.05)\}$ can not become an upperward 100α percentile minimum schedule for $\alpha \geq 0.05$. Therefore, these schedules should not be selected as a preferred schedule. If the decision maker is more conservative, the lower limit of α can be decreased down to any value less than 0.05. Any schedule in $\mathcal{N}_c(\alpha_1) \setminus \pi(\alpha_1)$ for the specified α_1 should be rejected from the set of candidates for a preferred schedule.

The inner frame drawn by the dotted line in Figure 1 for $\alpha = 0.2$ shows that any schedule in $\mathcal{N} \setminus \mathcal{N}_c(0.2)$ is dominated by $\pi(0.2)$ for $\alpha \leq 0.2$. If the decision maker is not so serious that the larger variance of total flow time is permitted, the upper limit of α can be increased to any value on [0.2, 0.5]. In this case, any schedule in $\mathcal{N} \setminus \mathcal{N}_c(\alpha_2)$ for the specified α_2 should be also rejected. Finally, we obtain a set of candidate schedules for a preferred schedule as $\{\mathcal{N}_c(\alpha_2) \setminus \mathcal{N}_c(\alpha_1)\} \cup \{\pi(\alpha_1)\}$ where α_1 and α_2 denote the lower and upper limits of α described above, respectively. However, in this set, there are still some schedules which cannot become the upperward 100α percentile minimum schedule for $\alpha \in [\alpha_1, \alpha_2]$, that is, such schedules become dominated for some values of $\alpha \in [\alpha_1, \alpha_2]$. Therefore, $\{\pi(\alpha) \mid \alpha_1 \leq \alpha \leq \alpha_2\}$ should be the final set of candidate schedules from which a preferred schedule will be selected.

If the number of elements included in this set is too small(or large), we recommend the decision maker to widen(or narrow) the range of α. After adjusting the size of the set $\{\pi(\alpha) \mid \alpha_1 \leq \alpha \leq \alpha_2\}$, the decision maker can select the preferred schedule from among this set of candidate schedules by comparing the criterion vector values of all candidate schedules each other. This final selection process is not difficult because only a few candidates remain at this stage.

We call this interactive system for selecting a preferred schedule the "interactive stochastic multiobjective scheduling system (ISMSS)," which is given by the following algorithm.

<Interactive Selection Algorithm For Single Machine Case>

<Step1> Calculate the sets of nondominated schedules $\mathcal{N}$, using the heuristic algorithm[9], and generate the sets of $\mathcal{N}_c(\alpha)$ and $\mathcal{N}^\alpha$ from $\mathcal{N}$. Plot them on a computer display as shown in Figure 1.

<Step2> Specify the upper and lower limits of α, $\alpha_1 \le \alpha \le \alpha_2$, as $\alpha_1 = 0.05$ and $\alpha_2 = 0.2$ tentatively. Draw the frames corresponding to these limits as shown in Figure 1.

<Step3> If necessary, adjust the upper or lower limit of α so that there a reasonable number of candidate schedules within the frames and so that the range of the upperward 100α percentile corresponding to $[\alpha_1, \alpha_2]$ is appropriate(satisfiable for the decision maker).

<Step4> Enlarge the figure expressing the set of $\{\pi(\alpha) \mid \alpha_1 \le \alpha \le \alpha_2\}$. Select a preferred schedule by comparing the vector values of candidate schedules each other. If necessary, return to Step 3 to adjust the limit frames of α.

This algorithm has been coded in the visual-C language. The sample color screen on a computer display will be shown in the following section for a parallel machine case.

4. Extension to Parallel Machine Case

In this section, we extend the ISMSS for a single-machine case to a parallel-machine case. In a parallel-machine case, all jobs should be assigned among machines. Therefore, we consider the following two cases: (1) the job assignment is given in advance and (2) the job assignment is not fixed. In case (1), if we evaluate every schedule for individual machine, the ISMSS discussed in the previous section can be directly applied to this case, called the " individual evaluation." This individual evaluation gives a good performance for each machine but does not always give a good result for an "overall evaluation" of all machines from the view point of whole system. Therefore, it may be better to evaluate each schedule from the latter point of view. In case (2), the individual evaluation of each machine becomes so complicated that the ISMSS is extended for only implementing the overall evaluation of all machines.

4.1. Fixed job assignment case

4.1.1. individual evaluation

In parallel-machine case with given job assignment, the individual two-objective scheduling problem for machine i can be formulated as follows:

$$\text{P7}^{(i)}\text{: } \min\left\{ \begin{pmatrix} E[F(\pi^{(i)})] \\ V[F(\pi^{(i)})] \end{pmatrix} \,\middle|\, \pi^{(i)} \in \Pi^{(i)}(S^{(i)}),\ S^{(i)} \text{ is given} \right\},\ i = 1 \sim M, \tag{4.1}$$

where $S^{(i)}$ denotes the set of jobs already assigned to machine i and $\Pi^{(i)}(S^{(i)})$ denotes the set of feasible schedules $\pi^{(i)}$ generated from $S^{(i)}$. Since the assignment $(S^{(1)}, S^{(2)}, \cdots, S^{(M)})$ is prespecified in this case, the set of nondominated schedules to each machine, denoted by $\mathcal{N}^{(i)}, i = 1 \sim M$, can be generated by applying the heuristic method (Jung et al.[9]) to the above individual problem.

4.1.2. overall evaluation

When we want to evaluate the overall performance in parallel-machine case, the overall two-objective scheduling problem to minimize both the expected value and the variance of the total flow time of all jobs over all machines can be formulated as follows:

$$\text{P8: } \min\left\{ \begin{pmatrix} \sum_{i=1}^{M} E[F(\pi^{(i)})] \\ \sum_{i=1}^{M} V[F(\pi^{(i)})] \end{pmatrix} \,\middle|\, \pi^{(i)} \in \Pi^{(i)}(S^{(i)}),\ S^{(i)} \text{ is given},\ i = 1 \sim M \right\}. \tag{4.2}$$

Since any combination of $\pi^{(i)}, i = 1 \sim M$, is feasible to P8, the number of feasible schedules, denoted by $\pi = (\pi^{(1)}, \pi^{(2)}, \ldots, \pi^{(M)})$, becomes too large to search. Fortunately, dominated schedules to $\mathrm{P7}^{(i)}, i = 1 \sim M$, can not become any component of nondominated schedules to P8. Therefore, P8 can be equivalently converted into the following:

$$\text{P9: } \min\left\{ \begin{pmatrix} \sum_{i=1}^{M} E[F(\pi^{(i)})] \\ \sum_{i=1}^{M} V[F(\pi^{(i)})] \end{pmatrix} \,\middle|\, \pi^{(i)} \in \mathcal{N}^{(i)}(S^{(i)}),\ S^{(i)} \text{ is given, } i = 1 \sim M \right\}. \tag{4.3}$$

Since the objective functions and constraints in P9 are all additive, P9 can be decomposed as follows:

$$\text{P9}'\text{: } \min\left\{ \min\left\{ \begin{pmatrix} \sum_{i=1}^{M-1} E[F(\pi^{(i)})] \\ \sum_{i=1}^{M-1} V[F(\pi^{(i)})] \end{pmatrix} \,\middle|\, \pi^{(i)} \in \mathcal{N}^{(i)},\ i = 1 \sim M-1 \right\} + \begin{pmatrix} E[F(\pi^{(M)})] \\ V[F(\pi^{(M)})] \end{pmatrix} \,\middle|\, \pi^{(M)} \in \mathcal{N}^{(M)} \right\}. \tag{4.4}$$

This decomposition form implies that the original P9 can be reduced into some small-sized problems, being easy to solve. The overall evaluation algorithm for obtaining the set of nondominated schedules to P9 is composed of the following four phases: (1) Find a set of nondominated schedules to $\mathrm{P7}^{(i)}, i = 1 \sim M$, individually; (2) Generate a set of feasible candidate schedules for a two-parallel machine problem, providing the set $\{\pi \mid \pi = (\pi^{(1)}, \pi^{(2)}), \forall\pi^{(1)} \in \mathcal{N}^{(1)}, \forall\pi^{(2)} \in \mathcal{N}^{(2)}\}$. Find a set of nondominated schedules for the set of candidate schedules with respect to the two objectives of $\sum_{i=1}^{2} E[F(\pi^{(i)})]$ and $\sum_{i=1}^{2} V[F(\pi^{(i)})]$; (3) Let $\mathcal{N}'$ be the set of nondominated schedules to a $(k-1)$-parallel machine problem. Find the set of nondominated schedules to the following k-parallel machine problem in a similar way to phase (2):

$$\min\left\{ \begin{pmatrix} E[F(\pi') + E[F(\pi^{(k)})] \\ V[F(\pi'] + V[F(\pi^{(k)})] \end{pmatrix} \,\middle|\, \pi' \in \mathcal{N}',\ \pi^{(k)} \in \mathcal{N}^{(k)} \right\}.$$

(4) Repeat phase (3) until the overall problem has been solved, that is, until the case of $k = M$ in phase (3) has been finished.

The algorithm to generate the set of nondominated schedules in this case is given as follows:

<Generation algorithm for overall evaluation in fixed job assignment case>

```
<Step1> Generate the set of nondominated schedules to $\mathrm{P7}^{(i)}$, $\mathcal{N}^{(i)}$, $i = 1 \sim M$.
<Step2> Arrange schedules $\pi_l^{(i)}$, $l = 1 \sim L^{(i)}$, in $\mathcal{N}^{(i)}$ in order of increasing variance and
        set $\mathcal{N} = \mathcal{N}^{(1)}$.
<Step3> Set $i = 1$.
<Step4> If $i = M$, stop (the current $\mathcal{N}$ is the set of nondominated schedules for overall
        evaluation).
<Step5> Set $\mathcal{N}' = \mathcal{N}$ and $i = i + 1$, and let $L'$ be the number of elements in $\mathcal{N}'$ so that
        $\mathcal{N}' \equiv \{\pi'_1, \pi'_2, \cdots, \pi'_{L'}\}$.
<Step6> Set $l = 1$.
<Step7> Set $\widetilde{\mathcal{N}} = \{\tilde{\pi}_s \mid \tilde{\pi}_s = (\pi'_l, \pi_s^{(i)}),\ s = 1 \sim L^{(i)}\}$
<Step8> If $l = L'$, set $\mathcal{N} = \widetilde{\mathcal{N}}$ and return to Step 4. Otherwise, set $l = l + 1$.
<Step9> Set $\overline{\mathcal{N}} = \{\bar{\pi}_s \mid \bar{\pi}_s = (\pi'_l, \pi_s^{(i)}),\ s = 1 \sim L^{(i)}\}$.
<Step10> Comparing the elements of $\overline{\mathcal{N}}$ with $\widetilde{\mathcal{N}}$ in order of $\bar{\pi}_1, \bar{\pi}_2, \ldots, \bar{\pi}_{L^{(i)}}$, find the new
        set of nondominated schedules $\mathcal{N}$ for $\widetilde{\mathcal{N}} \cup \overline{\mathcal{N}}$.
<Step11> Return to Step 8
```

4.2. General parallel-machine case

In general parallel-machine case without fixed job assignment, we only consider the overall evaluation as mentioned above. The problem can be formulated as follows:

$$\text{P10: min}\quad \left\{ \begin{pmatrix} \sum_{i=1}^{M} E[F(\pi^{(i)})] \\ \sum_{i=1}^{M} V[F(\pi^{(i)})] \end{pmatrix} \middle| \, \pi^{(i)} \in \Pi^{(i)}(S^{(i)}),\ i = 1 \sim M,\ \bigcup_{i=1}^{M} S^{(i)} = S \right\}, \tag{4.5}$$

where S denotes a set of jobs to be assigned among M machines and $S^{(i)}, i = 1 \sim M$, represents any decomposition of S such that $\bigcup_{i=1}^{M} S^{(i)} = S$ and $\bigcap_{i=1}^{M} S^{(i)} = \phi$, $i = 1 \sim M$.

Ho et al.[5] developed an efficient algorithm to minimize the number of tardy jobs in parallel-machine scheduling and Cheng et al.[2] reviewed parallel-machine scheduling methods. However, we cannot directly apply their methods to our case which has two objective functions including non-additive form. Instead, we reduce the set of feasible schedules to be searched below.

For any decomposition $S^{(i)}, i = 1 \sim M$, any combination of $\pi^{(i)} \in \Pi^{(i)}$, $i = 1 \sim M$, can be feasible to P10 but all dominated schedules in $\Pi^{(i)}$can be removed as well as in P8. Therefore, P10 can be reduced into the following:

$$\text{P11: min}\quad \left\{ \begin{pmatrix} \sum_{i=1}^{M} E[F(\pi^{(i)})] \\ \sum_{i=1}^{M} V[F(\pi^{(i)})] \end{pmatrix} \middle| \, \pi^{(i)} \in \mathcal{N}^{(i)}(S^{(i)}),\ i = 1 \sim M,\ \bigcup_{i=1}^{M} S^{(i)} = S \right\}. \tag{4.6}$$

Once a decomposition $(S^{(1)}, S^{(2)}, \cdots, S^{(M)})$ is given, P11 can be transformed into P9 and the set of nondominated schedules to P11 for the decomposition can be derived by using the same four phase method as that in the fixed job assignment case. A final set of nondominated schedules to P11 can be obtained by evaluating the whole sets of nondominated schedules generated from all possible decompositions of S. However, the number of possible decompositions of N jobs to be assigned among M machines is up to $M^N/M!$, requiring a lot of computation time to search. Fortunately, as shown in Theorem 1, it is sufficient to consider only balanced job assignments where the difference in the number of jobs assigned to each machine is one at most. This limitation for job assignments decreases the number of job assignments to search into $N!/\{(\lfloor N/M \rfloor !)^M M!\}$ approximately, where $\lfloor x \rfloor$ stands for a maximum integer less than or equal to x.

Theorem 1

Consider an N-job, M-parallel machine, stochastic, two-objective scheduling defined by problem P10(or P11). The whole set of nondominated schedules to P10(or P11) can be generated only by considering all decompositions $(S^{(1)}, S^{(2)}, \cdots, S^{(M)})$ satisfying $\bigcup_{i=1}^{M} S^{(i)} = S$ and

$$N_i = \begin{cases} \lfloor N/M \rfloor + 1, & i = 1 \sim N - \lfloor N/M \rfloor M; \\ \lfloor N/M \rfloor, & i = N - \lfloor N/M \rfloor M + 1 \sim M. \end{cases} \tag{4.7}$$

where $N_i \equiv |S^{(i)}|$, $i = 1, 2, \cdots, M$.

(proof)

Consider first a parallel two-machine case, where N_1 and N_2 jobs are assigned to machine M_1 and M_2, respectively. Suppose the balance in the number of jobs assigned to these machines is not good, that is, $N_1 > N_2 + 1$. Letting $c_{[i]j}$ be processing time of job $J_{[i]}$ scheduled at the ith position in machine M_j, we obtain the following relationships:

$$TF = \sum_{i=1}^{N_1} (N_1 - i + 1) c_{[i]1} + \sum_{j=1}^{N_2} (N_2 - j + 1) c_{[j]2}$$

$$\begin{aligned}
&= \textstyle\sum_{i=2}^{N_1}(N_1 - i + 1)c_{[i]1} + N_1 c_{[1]1} + \sum_{j=1}^{N_2}(N_2 - j + 1)c_{[j]2} \\
&= \textstyle\sum_{i=1}^{N_1-1}((N_1 - 1) - i + 1)c_{[i+1]1} + \sum_{j=1}^{N_2+1}((N_2 + 1) - j + 1)c_{[j-1]2} \\
&\quad +(N_1 - (N_2 + 1))c_{[1]1} \\
&> \textstyle\sum_{i=1}^{N_1-1}((N_1 - 1) - i + 1)c_{[i+1]1} + \sum_{j=1}^{N_2+1}((N_2 + 1) - j + 1)c_{[j-1]2}, \qquad (4.8)
\end{aligned}$$

where TF means the total flow time summed over two machines and $c_{[1]1} = c_{[0]2}$. This inequality shows that removing a job scheduled at the first position in machine M_1(the machine with more jobs) and adding it to the first position in machine M_2(the machine with fewer jobs) can decrease the total flow time TF, and can also decrease the expected value and the variance of total flow time. Therefore, it is sufficient to consider a set of job assignment satisfying $N_2 = \lfloor N/2 \rfloor$ and $N_1 = N - N_2$.

This principle also holds in any pair of two machines for an $M(> 2)$-parallel machine case, that is, it is sufficient to consider a set of job assignments in which the number of jobs assigned to each machine differs by one at most.

(Q.E.D.)

Even after the number of jobs to be assigned among machines is specified by Eq.(4.7), there are still many decompositions of jobs, denoted by $\boldsymbol{S}_k = (S_k^{(1)}, S_k^{(2)}, \ldots, S_k^{(M)})$, $k = 1, 2, \ldots$. For some decompositions which can not provide any nondominated schedule to P10, we can omit the further searching procedure. Fortunately, we can calculate the minimum values of $\sum_{i=1}^{M} E[F(\pi^{(i)})]$ and $\sum_{i=1}^{M} V[F(\pi^{(i)})]$ for any $\boldsymbol{S}_k$ by scheduling jobs in order of mean processing time and in order of variance of processing time in each machine, respectively. For a given $\boldsymbol{S}_k$, let $\boldsymbol{e}_k$ and $\boldsymbol{v}_k$ be the criterion vector of the point with $\min \sum_{i=1}^{M} E[F(\pi^{(i)})]$ and that of the point with min $\sum_{\rm i=1}^{\rm M} \rm V[F(\pi^{(i)})]$, respectively. The ideal point IP_k whose coordinate is given by (the minimum expected total flow time, the minimum variance of total flow time), is obtained for $\boldsymbol{S}_k$ by using the coordinates of $\boldsymbol{e}_k$ and $\boldsymbol{v}_k$. If there exists either $\boldsymbol{e}_j$ or $\boldsymbol{v}_j$ for $\boldsymbol{S}_j$ dominating IP_k, any schedule generated from $\boldsymbol{S}_k$ can not become nondominated for P10. In this case, we call "decomposition $\boldsymbol{S}_k$ is dominated by $\boldsymbol{S}_j$." It is better to remove such dominated decompositions from the set of candidates to search in advance.

From this point of view, we propose an algorithm to generate the set of nondominated schedules in a general parallel-machine case as follows:

<Generation algorithm for overall evaluation in general parallel machine case>

```
<Step1> Determine the number of jobs to be assigned among machines, $N_i$, $i = 1 \sim M$,
    according to Eq.(4.7).
<Step2> Set $k = 1, D = \{1\}$ and $EV = \phi$, and generate the $k-$th assignment, $S_k^{(i)}, i =
    1 \sim M$.
<Step3> Calculate $\boldsymbol{e}_k, \boldsymbol{v}_k$, and $IP_k$ for $S_k^{(i)}, i = 1 \sim M$.
<Step4> If $IP_k$ is dominated by some element in $EV$, go to Step 8
<Step5> Set $EV = EV \cup \{\boldsymbol{e}_k, \boldsymbol{v}_k\}$ and eliminate any element in the new $EV$ dominated
    by some element in the new $EV$.
<Step6> If there is $l$ in $D$, such that $IP_l$ is dominated by some element in $EV$, delete all
    such $l$ from $D$.
<Step7> Set $D = D \cup \{k\}$.
<Step8> If all decompositions are generated, go to Step 9. Otherwise, generate a new
    decomposition, set $k = k + 1$ and return to Step 3.
```

<Step9> Number the elements in D in order of variance of total flow time of $\boldsymbol{v}_k$, from 1 to K, where K is the number of elements in D.
<Step10> Set $k=1$, and $\mathcal{N}=\phi$.
<Step11> Generate the set of nondominated schedules $\mathcal{N}_k$ for $S_k^{(i)}$, $i=1 \sim M$, by using the generation algorithm for overall evaluation in the fixed job assignment case.
<Step12> If there are dominated elements in $\mathcal{N} \cup \mathcal{N}_k$, delete them, and rename it $\mathcal{N}$.
<Step13> If $k<K$, set $k=k+1$ and return to Step 11.
<Step14> Stop(the current $\mathcal{N}$ is the set of nondominated schedules).

4.3. Selection of the preferred schedule

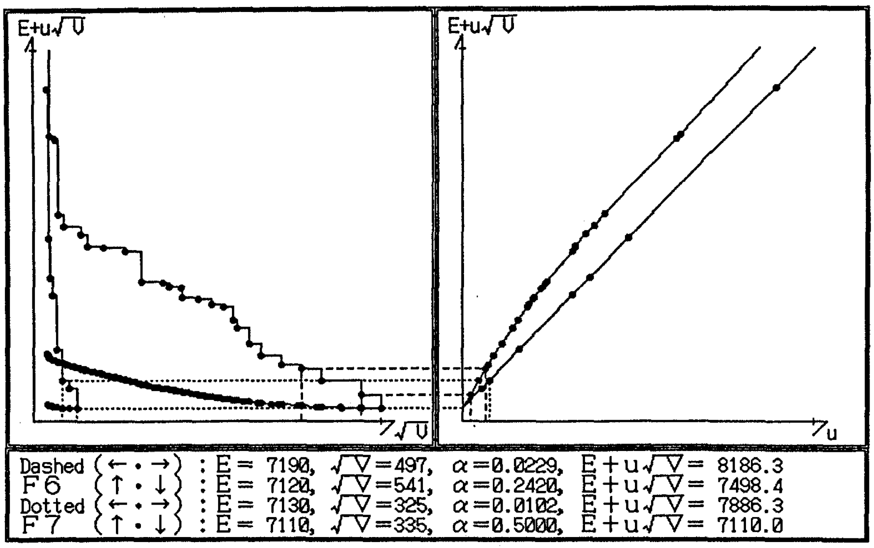
The basic idea for selecting the preferred schedule in the parallel-machine case is the same as that in the single machine case. Especially for the overall evaluation, the selection method in the single machine case can be used without any change. On the other hand, the individual evaluation in the fixed job assignment case requires a slight change as mentioned below.

We show an example of two-parallel machine case whose input data are given by patterns 1 and 2 in Table 1 for machines M_1 and M_2, respectively. The sets of nondominated schedules to machines M_1 and M_2 are shown in Tables 2 and 3, respectively. Figure 2 is a computer display expressing the set of nondominated schedules to the parallel-machine case and related information as shown in Figure 1 for the single-machine case. At the bottom of the left-hand-side in Figure 2, two series of ● (blue on an actual computer display) denote the sets of nondominated schedules: the lower series is for machine M_1 and the upper series is for machine M_2. Both sets have the same minimum expected value of total flow time and the same minimum variance of total flow time, but the range and shape of each set is different from each other. The two dotted lines in Figure 2 are the limit frames of α for machine M_1 and the two dashed lines, for machine M_2. The outer and inner limit frames of α represent the lower and upper limits of α for each machine, respectively. In the space below the graph in Figure 2, the expected value of total flow time(E), the standard deviation of total flow time$(\sqrt{V})$, the value of α and the upperward 100α percentile$(E+u\sqrt{V})$ are shown for the nondominated schedules corresponding to the lower and upper limits of α. The first two rows in this space correspond to machine M_2 and the last two rows, to machine M_1. "F7(or F6)" in this space means that after pushing "F7(or F6) function key" we can do operations related to machine M_1(or M_2). The arrow keys "←" and "→" are used for moving up and down the outer frame of α for each machine, respectively. The arrow keys "↑" and "↓" are also used for moving up and down the inner frame of α for each machine, respectively. On this display, we can move the outer and inner limit frames of α by pushing corresponding "arrow key" on a key board individually after pushing either "F7 function key" for machine M_1 or "F6 function key" for machine M_2. These frames should be moved so that the number of nondominated schedules within the upper and lower frames should be sufficiently small and so that the range of α should be similar to each other between two machines. In this case, the resulting frames are selected as $0.0102 \leq \alpha \leq 0.5000$ including three nondominated schedules for machine M_1 and $0.0229 \leq \alpha \leq 0.2420$ including eight nondominated schedules for machine M_2.

After pushing "F6 function key" and "F2 function key" successively, the enlarged display for machine M_2 appears as shown in Figure 3. At the bottom of the left-hand-side on this display, eight nondominated schedules within the limit frames of α are expressed by a series of ● (blue on an actual computer display), and three upperward 100α percentile minimum schedules are expressed by a series of ● (red on an actual computer display) at the top on the left-hand-side and also on the right-hand-side of the same display. If there are some

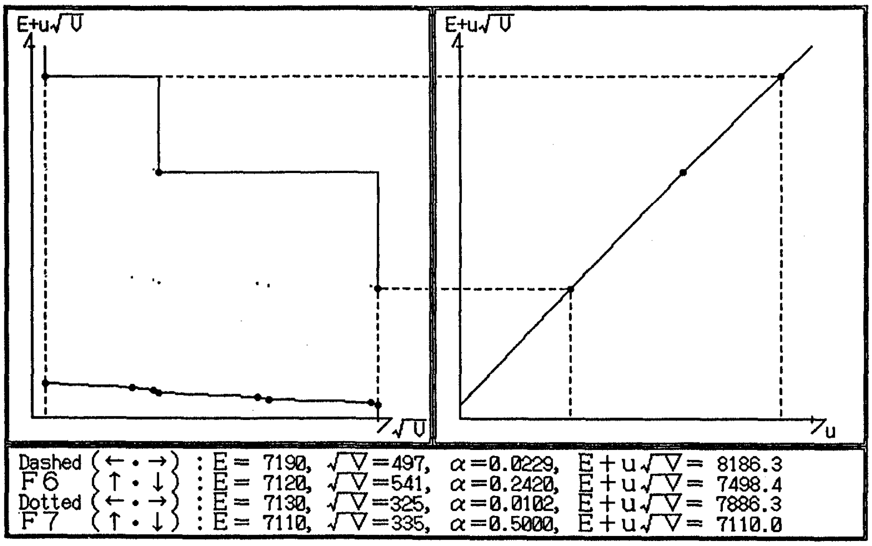
Table 3. The set of nondominated schedules for the single-machine case (pattern 2: "*" denotes the element of $\mathcal{N}^\alpha$ and $\sqrt{V[F]}$ denotes $\sqrt{V[F(\pi)]}$)

No.	$E[F(\pi)]$	$\sqrt{V[F]}$	$u_\alpha(\pi)$	No.	$E[F(\pi)]$	$\sqrt{V[F]}$	$u_\alpha(\pi)$	No.	$E[F(\pi)]$	$\sqrt{V[F]}$	$u_\alpha(\pi)$
1*	7110	555	0.70	43	7530	417	5.53	85	7990	359	8.39
2*	7120	541	1.39	44	7540	416	4.96	86	8000	358	7.31
3	7130	540	0.74	45	7550	414	3.87	87	8010	357	6.40
4	7140	527	1.38	46*	7560	412	6.76	88	8020	356	5.68
5	7150	525	0.76	47*	7570	410	6.83	89*	8030	354	9.68
6*	7160	512	2.00	48	7580	409	5.05	90	8050	354	7.04
7	7170	511	1.41	49	7590	407	6.63	91	8060	353	6.10
8	7180	509	0.87	50	7600	406	6.61	92	8070	352	3.74
9*	7190	497	2.19	51	7610	405	4.24	93	8080	349	9.23
10	7200	495	1.70	52	7620	403	6.61	94	8100	348	8.31
11	7210	494	0.91	53*	7630	401	7.10	95	8110	347	7.08
12*	7220	483	2.74	54	7640	401	4.92	96	8130	346	3.94
13	7230	481	2.37	55	7650	399	6.54	97*	8140	343	10.6
14	7240	481	1.69	56*	7660	397	7.16	98	8160	341	10.0
15	7250	475	1.66	57	7670	396	6.50	99	8180	341	4.08
16*	7260	469	3.38	58	7680	395	6.98	100*	8190	338	11.2
17	7270	467	2.92	59	7700	392	6.43	101	8210	337	11.2
18	7280	466	1.68	60	7710	390	7.08	102	8220	336	7.72
19*	7290	460	4.32	61	7730	388	6.36	103	8230	334	11.2
20	7300	458	4.16	62	7740	386	6.83	104	8240	334	7.68
21	7310	457	3.17	63	7750	385	4.13	105	8250	333	11.2
22	7320	455	2.31	64	7760	383	6.29	106	8270	331	11.1
23*	7330	451	4.77	65*	7770	382	9.36	107	8290	329	11.0
24*	7340	449	5.58	66	7780	381	6.26	108	8300	329	6.04
25	7350	448	4.80	67	7790	380	9.22	109	8310	328	11.1
26	7360	446	4.72	68	7810	379	6.22	110	8320	327	7.26
27	7370	444	3.32	69	7820	377	8.19	111*	8330	326	12.2
28*	7380	441	5.77	70	7830	377	7.59	112	8340	325	10.8
29	7390	440	5.47	71	7840	375	7.53	113	8350	325	5.96
30	7400	438	5.27	72	7850	374	7.45	114	8360	324	11.0
31	7410	437	4.63	73	7860	373	6.12	115	8370	323	5.93
32	7420	436	3.27	74	7870	371	7.02	116*	8380	322	18.4
33*	7430	433	6.10	75*	7880	370	9.59	117	8400	321	10.9
34	7440	432	3.69	76	7900	370	6.70	118*	8420	320	18.7
35	7450	430	5.89	77	7910	368	7.74	119	8450	318	18.2
36	7460	428	5.56	78	7920	367	6.94	120*	8490	316	38.4
37	7470	427	4.88	79	7930	366	8.12	121*	8520	315	46.8
38	7480	426	3.25	80	7940	365	7.26	122	8540	315	38.3
39*	7490	423	6.20	81	7950	364	6.29	123	8560	314	25.9
40	7500	422	5.21	82	7960	363	3.81	124*	8570	314	58.5
41	7510	420	6.11	83	7970	361	9.53	125*	8610	313	∞
42	7520	418	5.93	84	7980	361	6.82				



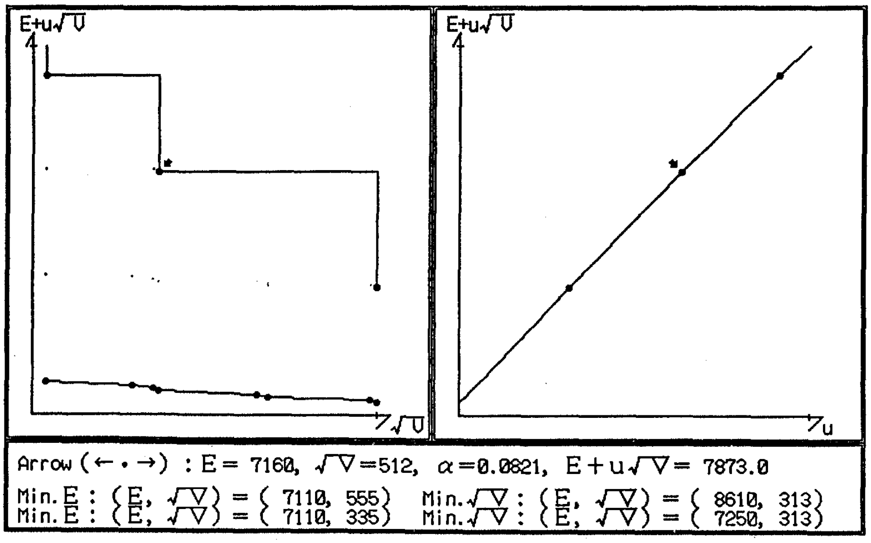
```
Dashed (← • →) : E= 7190, √V=497, α=0.0229, E+u√V= 8186.3
F6     (↑ • ↓) : E= 7120, √V=541, α=0.2420, E+u√V= 7498.4
Dotted (← • →) : E= 7130, √V=325, α=0.0102, E+u√V= 7886.3
F7     (↑ • ↓) : E= 7110, √V=335, α=0.5000, E+u√V= 7110.0
```

Figure 2. Computer display of the set of nondominated schedules in parallel-machine case



```
Dashed (← • →) : E= 7190, √V=497, α=0.0229, E+u√V= 8186.3
F6     (↑ • ↓) : E= 7120, √V=541, α=0.2420, E+u√V= 7498.4
Dotted (← • →) : E= 7130, √V=325, α=0.0102, E+u√V= 7886.3
F7     (↑ • ↓) : E= 7110, √V=335, α=0.5000, E+u√V= 7110.0
```

Figure 3. Enlargement of nondominated schedules within the limit frames of α



```
Arrow (← • →) : E= 7160, √V=512, α=0.0821, E+u√V= 7873.0
Min.E : (E, √V) = ( 7110, 555)  Min.√V : (E, √V) = ( 8610, 313)
Min.E : (E, √V) = ( 7110, 335)  Min.√V : (E, √V) = ( 7250, 313)
```

Figure 4. Final computer display for selecting a preferred schedule

upperward 100α percentile minimum schedules within the frame, a decision maker will not select the nondominated schedules which cannot become the upperward 100α percentile minimum schedule as a preferred schedule. Therefore, the candidate schedules for a preferred schedule are reduced into only three nondominated schedules at this searching stage.

From this stage, pushing "F3 function key," we proceed to the last stage for selecting a preferred schedule as shown in Figure 4. On this display, the bold arrow expresses one of candidate schedules and the corresponding data are given at the top of the bottom space, in which the second and third rows show the criterion vector values of the two minimum points with respect to the expected value and variance of total flow time at machines M_2 and M_1, respectively. Pushing "arrow key" moves the bold arrow on the display one by one. It is not difficult for the decision maker to select a preferred schedule comparing data with each other on the display, because the set of nondominated schedules are reduced into such a small set at this final stage.

If necessary, we can go back to the display immediately before the current one by pushing "F4 function key" and can go back to the initial display(Figure 2) by pushing "F1 function key."

The ISMSS algorithm for the individual evaluation in a fixed job assignment, parallel machine case is given as follows:

<Interactive selection algorithm for individual evaluation in parallel machine case>

<Step1> Calculate the set of nondominated schedules $\mathcal{N}^{(i)}$, $i = 1 \sim M$, applying the heuristic algorithm[9] to each problem defined by P7(i), $i = 1 \sim M$. Generate the sets of $\mathcal{N}_c^{(i)}(\alpha)$ and $\mathcal{N}^{\alpha(i)}$ from $\mathcal{N}^{(i)}$, $i = 1 \sim M$, and plot them on a computer display as shown in Figure 2.

<Step2> Specify the upper and lower limits of α, $\alpha_1^{(i)} \leq \alpha \leq \alpha_2^{(i)}$, $i = 1 \sim M$, as $\alpha_1^{(i)} = 0.05$ and $\alpha_2^{(i)} = 0.2$, $i = 1 \sim M$, tentatively. Draw the frames corresponding to these limits as shown in Figure 2.

<Step3> If necessary, adjust the upper or lower limit of α individually so that there are a reasonable number of candidate schedules within the frames in each machine, and so that the range of the upperward 100α percentile corresponding to $[\alpha_1^{(i)}, \alpha_2^{(i)}]$, $i = 1 \sim M$, is appropriate and not far different from each other.

<Step4> Enlarge the figure expressing the set of $\{\pi^{(i)}(\alpha) \mid \alpha_1^{(i)} \leq \alpha \leq \alpha_2^{(i)}\}$, $i = 1 \sim M$, as shown in Figure 3. Select a preferred schedule by comparing the criterion vector values of candidate schedules displayed in Figure 4. If necessary, return to Steps 2 and 3 to adjust the limit frames of α.

The ISMSS algorithm for the overall evaluation in the parallel machine case is the same as that for a single machine case with $\mathcal{N}$ obtained by the proposed generation algorithm for overall evaluation.

5. Conclusions

How to select a preferred schedule from among a lot of nondominated schedules is one of difficult problems in multiobjective scheduling. This paper proposed an interactive decision system, called " ISMSS," to select a preferred schedule in the Stochastic Multiobjective Scheduling Problem(SMSP) with random processing times. Since the relationships between the expected value and the variance of the total flow time can be represented by the upperward 100α percentile, some new categories of nondominated schedules were presented and used in the ISMSS, where the stochastic meaning of the upperward 100α percentile helps shop-floor managers select the preferred schedule from among a lot of nondominated schedules. In parallel-machine case, a heuristic method was proposed for generating the set of nondominated schedules on the basis of the heuristic algorithm proposed by Jung et al., and the ISMSS for a single-machine case was extended to the parallel-machine case. Some numerical examples were shown to demonstrate the effectiveness of the proposed system. This method can be widely applied to the other stochastic multiobjective problems.

The algorithm proposed for the general parallel machine case is not elaborated so well that the case with a lot of nondominated schedules cannot be solved not only because of the computation time problem but also because of the memory size problem. However, in a real situation, it is not necessary to generate the exact set of nondominated schedules because many schedules with the same values of performance measures are generated in the set of nondominated schedules for the general parallel machine case. Therefore, we can avoid an unnecessary effort for generating such schedules by removing some kinds of job decompositions to search, saving a lot of computation time and memory capacity. This task remains as one of our future works.

References

[1] F. B. Abdelaziz, P. Lang and R. Nadeau: Distributional Efficiency in Multiobjective Stochastic Linear Programming. *European Journal of Operational Research*, **85**(1995) 399-415.

[2] T. C. E. Cheng and C. C. S. Sin: A State-of-the-Art Review of Parallel-Machine Scheduling Research. *European Journal of Operational Research*, **47**(1990) 271-292.

[3] R. L. Daniel and R. J. Chambers: Multiobjective Flow-Shop Scheduling. *Naval Research Logistics*, **37**(1990) 981-995.

[4] F. G. Forst: Bicriterion Stochastic Scheduling on One or More Machines. *European Journal of Operational Research*, **80**(1995) 404-409.

[5] J. C. Ho and Y. Chang: Minimizing the Number of Tardy Jobs for m Parallel Machines. *European Journal of Operational Research*, **84**(1995) 343-355.

[6] K. Huckert and R. O. Roglin: On the Interactive Solution to a Multicriteria Scheduling Problem. *Zeitshrift für Operations Research*, **24**(1980) 47-60.

[7] H. Ishii and M. Tada: Single Machine Scheduling Problem with Fuzzy Precedence Relation. *European Journal of Operational Research*, **87**(1995) 284-288.

[8] R. R. Joshi: A New Approach to Stochastic Programming Problems: Discrete Model. *European Journal of Operational Research*, **83**(1995) 514-529.

[9] Y. S. Jung, H. Nagasawa and N. Nishiyama: Bicriterion Single-Stage Scheduling to Minimize Both the Expected Value and the Variance of Total Flow Time. *Journal of Japan Industrial Management Association,* **39**(1990) 76-82, in Japanese.

[10] H. Kise, A. Shiomi, M. Uno and D. Chao: An Efficient Algorithm for a Chance-Constrained Scheduling Problem. *Journal of the Operations Research Society of Japan*, **25**(1982) 193-204.

[11] P. Korhonen, H. Moskowitz and J. Wallenius: Multiple Criteria Decision Support – A Review. *European Journal of Operational Research*, **63**(1992) 361-375.

[12] K. Morizawa, T. Ono, H. Nagasawa and N. Nishiyama: An Interactive Approach for searching a Preferred schedule. *Journal of Japan Industrial Management Association,* **39**(1993) 76-82, in Japanese.

[13] T. Murata, H. Ishibuchi and H. Tanaka: Multiobjective Genetic Algorithm and Its Applications to Flowshop Scheduling. *Computers industrial Engineering*, **30**(1996) 957-968.

[14] R. T. Nelson, R. K. Sarin and R. L. Daniels: Scheduling with Multiple Performance Measures; One Machine Case. *Management Science*, **32**(1986) 464-479.

[15] N. Nishiyama and H. Nagasawa: Design on Assembly Line with Distributed Operation Times(1st Report). *Transactions of the Japan Society of Mechanical Engineering*, **46C**(1980) 450-457, in Japanese.

[16] C. Rajendran: Two-Stage Flowshop Scheduling Problem with Bicriteria. *Journal of Operations Research Society*, **43**(1992) 871-884.

[17] C. Rajendran: Heuristics for Scheduling in Flowshop with Multiple Objectives. *European Journal of Operational Research*, **82**(1995) 540-555.

[18] T. Sen, F. M. E. Raiszadeh and P. Dileepan: A Branch-and-Bound Approach to the Bicriterion Scheduling Problem involving Flow Time and Range of Lateness. *Management Science*, **34**(1988) 254-260.

[19] W. E. Smith: Narious Optimizers for single Stage Production. *Naval Research Logistic Quarterly*, **3**(1956) 59-66.

[20] L. N. Van Wassenhove and L. F. Gelders: Solving a Bicriterion Scheduling Problem. *European Journal of Operational research,* **4**(1980) 42-48.

Hiroyuki Nagasawa
Department of Industrial Engineering
College of Engineering
Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka, 599-8531, Japan
E-mail: ng@ie.osakafu-u.ac.jp