REALIZATION OF AN OPTIMAL SCHEDULE FOR THE TWO-MACHINE FLOW-SHOP WITH INTERVAL JOB PROCESSING TIMES

Natalja Leshchenko, Yuri Sotskov

Abstract: Non-preemptive two-machine flow-shop scheduling problem with uncertain processing times of n jobs is studied. In an uncertain version of a scheduling problem, there may not exist a unique schedule that remains optimal for all possible realizations of the job processing times. We find necessary and sufficient conditions (Theorem 1) when there exists a dominant permutation that is optimal for all possible realizations of the job processing times. Our computational studies show the percentage of the problems solvable under these conditions for the cases of randomly generated instances with $n \le 100$. We also show how to use additional information about the processing times of the completed jobs during optimal realization of a schedule (Theorems 2 – 4). Computational studies for randomly generated instances with $n \le 50$ show the percentage of the two-machine flow-shop scheduling problems solvable under the sufficient conditions given in Theorems 2 – 4.

Keywords: Scheduling, flow-shop, makespan, uncertainty.

ACM Classification Keywords: F.2.2 Non-numerical algorithms and problems: Sequencing and scheduling

Introduction

In scheduling theory, it is usually assumed that the job processing times are known exactly before scheduling. However, the real-world scheduling problems usually are not deterministic: Machines may break down, activities may take longer time to be executed than it is expected before scheduling, jobs may be added or canceled, etc. In operations research literature, there are different approaches concerning management of uncertainty in scheduling (see surveys [Aytug et al., 2005; Davenport, Beck, 2000; Gupta, Stafford, 2006]).

The *stochastic method* [Elmaghraby, Thoney, 2000; Pinedo, 1995] for dealing with uncertainty is useful when the process has enough prior information to characterize the probability distributions of the random processing times and there are a lot of realizations of a similar process. In the particular case of the stochastic scheduling problem, random processing times may be *controllable*, and the objective is to choose the optimal processing times (which are under control of a decision-maker) and the optimal schedule with the chosen job processing times. For such a problem, the objective function depends on both the job processing times and the job completion times (see, e.g., [Jansen, Mastrolilli, Solis-Oba, 2005]). The current trends in the field of scheduling under the *fuzziness* notion have been presented in [Slowinski, Hapke, 1999]. In the field of operations research for the problems under uncertainty auxiliary criteria are often used. The most popular auxiliary criteria are criteria introduced by Wald, Hurwicz, and Savage (see [Shafransky, 2005] for a brief survey).

In spite of several developments, flow-shop scheduling problem with uncertain job processing times remains unsolved (see [Gupta, Stafford, 2006]). In the most of these developments, Johnson's rule and analysis methods play a significant role. In this paper, we consider a two-machine flow-shop scheduling problem with interval job processing times. A scheduling problem with interval job processing times is rather general, since most events that are uncertain before scheduling may be considered as factors that vary the job processing times. The processing time may depend on the distance between machines, the type of transport used, traffic conditions, intervals of availability of machines, possible machine breakdowns, emergence of new unexpected jobs with high priority, early or late arrival of raw materials, etc. In [Gupta, Stafford, 2006], there were discussed 21 restrictions involved in the classical flow-shop problem (denoted as $F \,||\, C_{\max}$) with the fixed job processing times, where criterion $C_{\max}$ denotes minimization of a schedule length. Nine of these restrictions addressed the criterion and type of the processing system, but all the remaining restrictions may be overcome by using suitable intervals for possible variations of the job processing times.

Problem Setting

We consider the non-preemptive flow-shop scheduling problem with two machines and random bounded job processing times (only lower and upper bounds of the job processing times are assumed to be given before scheduling). This problem is denoted as $F2 \mid t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \mid C_{\max}$.

Two machines $M = \{M_1, M_2\}$ have to process set of *n* jobs $J = \{1, 2, ..., n\}$ with the same machine route (M_1, M_2). All the *n* jobs are available to be processed from time $\tau = 0$. In contrast to deterministic scheduling problem, it is assumed that processing time t_{jm} of job $j \in J$ on machine $M_m \in M$ is not fixed before scheduling. In a realization of the process, t_{jm} may be equal to any real value between lower bound $t_{jm}^{L} \ge 0$ and upper bound $t_{jm}^{U} \ge t_{jm}^{L}$ being given before scheduling. The probability distribution of the random job processing time is unknown.

Thus, we address the stochastic flow-shop scheduling problem for the case when it is hard to obtain exact probability distributions for random bounded job processing times, and when assuming a specific probability distribution is not realistic. It has been observed that, although the exact probability distribution of the job processing times may be unknown in advance, upper and lower bounds on the job processing times are easy to obtain in many practical cases. In such a case there may not exist a unique schedule that remains optimal for all possible realizations of the job processing times and this question is considered in detail in the next section.

If equality $t_{jm}^{L} = t_{jm}^{U}$ holds for each job $j \in J$ and each machine $M_m \in M$, then problem $F2 \mid t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \mid C_{\max}$ turns into a *deterministic* flow-shop problem (denoted as $F2 \,||\, C_{\max}$) that is polynomially solvable due to Johnson's algorithm [Johnson, 1954]. Permutation that is constructed by Johnson's algorithm is called a *Johnson's permutation*. At least one optimal permutation for problem $F2 \,||\, C_{\max}$ is a Johnson's permutation. (It should be noted however, that for the problem $F2 \,||\, C_{\max}$, an optimal schedule may also be defined by permutation that is not a Johnson's permutation.)

In contrast to deterministic problem $F2 \,||\, C_{\max}$ we call problem $F2 \mid t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \mid C_{\max}$ as an *uncertain* scheduling problem.

Existence of a Dominant Johnson's Permutation for the Uncertain Flow-Shop Problem

Let *T* denote a set of feasible vectors $t = (t_{1,1}, t_{1,2}, ..., t_{n1}, t_{n2})$ of the job processing times:

$$T = \left\{ t \mid t_{jm}^{L} \le t_{jm} \le t_{jm}^{U},\ j \in J, m \in M \right\}.$$

The set *S* of all feasible permutations (schedules) has cardinality $|S| = n!$. Permutation $\pi_i \in S$ *dominates* each other permutation $\pi_k \in S$ with $k \ne i$ if inequality $C_{\max}(\pi_i, t) \le C_{\max}(\pi_k, t)$ holds for each permutation $\pi_k \in S$, where $C_{\max}(\pi_i, t)$ denotes objective value $C_{\max}$ (length of a schedule) to the deterministic problem $F2 \,||\, C_{\max}$ with the fixed vector $t \in T$ of the job processing times.

We call permutation $\pi_i \in S$ a *dominant Johnson's permutation* to the uncertain problem $F2 \mid t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \mid C_{\max}$ if for any feasible vector $t \in T$ of the job processing times permutation π_i is a Johnson's permutation (so permutation π_i is optimal) for the deterministic problem $F2 \,||\, C_{\max}$ with this vector $t \in T$ of the job processing times.

We consider the case when inequality $t_{jm}^{L} < t_{jm}^{U}$ holds for each job $j \in J$ and each machine $M_m \in M$. For this case the following theorem has been proven in [Leshchenko, Sotskov, 2005].

Theorem 1 *Let* $t_{jm}^{L} < t_{jm}^{U}$, $j \in J$, $M_m \in M$. *Then there exists a dominant Johnson's permutation* $\pi_i \in S$ *to the uncertain problem* $F2 \,|\, t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \,|\, C_{\max}$ *if and only if:*

a) For any pair of jobs i and j with $t_{k1}^{U} \le t_{k2}^{L}$, *k = i, j (with* $t_{k2}^{U} \le t_{k1}^{L}$, *k = i, j, respectively) either* $t_{i1}^{U} \le t_{j1}^{L}$ *or* $t_{j1}^{U} \le t_{i1}^{L}$ *(either* $t_{i2}^{U} \le t_{j2}^{L}$ *or* $t_{j2}^{U} \le t_{i2}^{L}$*);*

b) There exists at most one job j^* *such that* $t_{j1}^{L} < t_{j2}^{U}, t_{j2}^{L} < t_{j1}^{U}$, *and the following inequalities hold:* $t_{j^*1}^{L} \ge \max\{t_{i1}^{U} \,|\, t_{i1}^{U} \le t_{i2}^{L}\}$, $t_{j^*2}^{L} \ge \max\{t_{i2}^{U} \,|\, t_{i2}^{U} \le t_{i1}^{L}\}$.

Computational Results for Necessary and Sufficient Conditions of Theorem 1

In this section, we consider randomly generated uncertain flow-shop problems $F2 \,|\, t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \,|\, C_{\max}$ and answer (by experiments on PC) the question of how many uncertain instances have a Johnson's permutation that is optimal for all corresponding deterministic problems $F2 \,||\, C_{\max}$ with feasible vectors $t \in T$ of the job processing times. Namely, for each randomly generated instance $F2 \,|\, t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \,|\, C_{\max}$ under consideration we tested whether condition of Theorem 1 hold.

The computational algorithm was coded in C++. For the experiments, we used an AMD 3000 MHz processor with 1024 MB main memory. For each job, the lower bound of job processing time was randomly generated in the range [1,1000] and the upper bound of job processing time was computed as follows: $t_{jm}^{U} = t_{jm}^{L}(1 + L\% / 100\%)$.

In each series we generated and tested 1000 instances (for each combination of *n* and *L* under consideration). Random instances have been generated as follows. We tested two-machine flow-shop scheduling problems with $n \in$ {5, 10, 15, …, 100} jobs, integer processing times uniformly distributed in the range [1,1000], and $L \in$ {1, 2, …, 10}.

Table 1. Percentage of instances solvable due to Theorem 1

L\n	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
1	94.5	85.1	70	53.5	36.9	24.9	14.7	8.2	4.8	1.9	1.3	0.6	0.3	0	0	0	0	0	0	0
2	91.2	69.3	45.6	24.2	11.8	4.2	1.1	0.6	0.1	0	0	0	0	0	0	0	0	0	0	0
3	87.7	58.4	28.3	10.3	3.8	1.3	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	82.4	47.4	18.8	5.5	0.8	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	75.5	37.4	11.3	2.2	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	72.5	32.4	7.8	1.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	66.9	25	5.4	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	65.8	19.9	2.1	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	63.4	18	2.4	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	59.2	14.4	2	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 1 presents the percentage of instances in the series for which conditions of Theorem 1 hold. From our experiment, it follows that increasing simultaneously both numbers *n* and *L* decreases the number of instances solvable due to Theorem 1.

Optimal Realization of a Schedule

In Table 1, there are many cases for which percentage of instances solvable due to Theorem 1 is equal to 0, i.e., unique dominant permutation does not exist in each of such cases. For such instances, we propose to use modified Johnson's algorithm developed in [Leshchenko, Sotskov, 2005] to construct partial strong order of jobs *J* (we considered only the cases when strong inequality $t^L_{jm} < t^U_{jm}$ holds for each job $j \in J$ and each machine $M_m \in M$).

Theorem 2 *Let inequality $t^L_{jm} < t^U_{jm}$ hold for each job $j \in J$ and each machine $M_m \in M$. The order $j \to w$ of jobs $j \in J$ and $w \in J$ is optimal for processing jobs of set J if and only if at least one of the following three conditions holds:*

$$t^U_{w2} \le t^L_{w1} \text{ and } t^U_{j1} \le t^L_{j2};$$

$$t^U_{j1} \le t^L_{w1} \text{ and } t^U_{j1} \le t^L_{j2};$$

$$t^U_{w2} \le t^L_{w1} \text{ and } t^U_{w2} \le t^L_{j2}.$$

Let us consider instances with graph of the above partial strong order when no more than two jobs are not in this order at any time moment (see Fig. 1 for example).

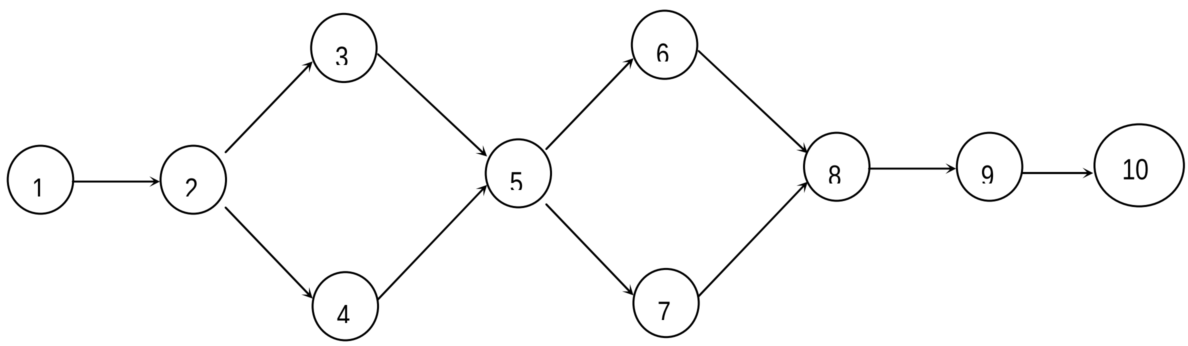

Figure 1. A graph reduction of the partial strong order constructed due to modified Johnson's algorithm

To answer the question of how many instances of the uncertain two-machine flow-shop scheduling problem have such form of the partial strong order (constructed due to Theorem 2), we made experiments on PC analogous to those presented in the previous section.

We call pair of jobs *conflict* pair of jobs if they are ready for processing but their optimal order is not fixed in the above partial strong order (see pair of jobs 3 and 4, and pair of jobs 6 and 7 in Fig. 1). Table 2 presents the percentage of instances in the series where no more than one conflict pair of jobs exists at any time. As we see from Table 2, increasing both numbers *n* and *L* decreases the number of instances with no more than one conflict pair of jobs existing at any time.

For each series we generated and tested 100 instances (for each combination of *n* and *L* under consideration). Random instances have been generated as follows. We tested problems with $n \in \{5, 10, 15, \ldots, 50\}$ jobs, integer job processing times uniformly distributed in the range [10,1000], and $L \in \{1, 2, 3, \ldots, 15\}$. For each job, the lower bound of the job processing times was randomly generated in the range [10,1000] and the upper bound of the job processing times was computed as follows: $t^U_{jm} = t^L_{jm}(1 + L\% / 100\%)$. We restricted the experiments by $n \le 50$ due to the results presented in Table 2.

Table 2. Percentage of instances for which no more than one conflict pair of jobs exists at any time (the lower bounds of the processing times are uniformly distributed in the range [10, 1000] with $t^U_{jm} = t^L_{jm}(1 + L\% / 100\%)$)

n\ L	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	5	8	8	16	18	25	26	20	25	29	37	40	38	36	46
10	15	29	28	37	51	50	54	52	52	63	60	45	55	53	42
15	32	44	48	59	63	55	48	40	45	37	38	33	33	19	23
20	41	65	47	48	45	47	34	36	22	17	13	6	10	5	4
25	52	63	60	42	21	18	14	8	9	8	6	5	2	1	1
30	67	53	50	20	20	12	8	3	3	0	0	0	0	0	0
35	68	51	33	24	10	2	1	0	0	0	0	0	0	0	0
40	69	34	22	5	4	0	0	0	0	0	0	0	0	0	0
45	57	35	18	9	1	0	0	0	0	0	0	0	0	0	0
50	50	21	10	0	0	0	0	0	0	0	0	0	0	0	0

For a conflict pair of jobs, the following sufficient conditions for constructing an optimal order for processing two jobs have been proven. If in the above partial strong order of jobs (with no more than one conflict pair of jobs at any time) two first jobs make conflict pair, then one can use the following sufficient conditions for optimal ordering of the conflict jobs.

Theorem 3 *Let jobs 1 and 2 make first conflict pair of jobs and let job 3 must be processed optimally after them. Then the order $1 \to 2$ of jobs $1 \in J$ and $2 \in J$ is optimal for processing jobs from set J if at least one of the following three conditions holds:*

$$t^U_{1,1} + t^U_{2,2} + \max\{t^U_{1,2}; t^U_{2,1}\} \le t^L_{1,2} + t^L_{2,1} + \max\{t^L_{1,1}; t^L_{2,2}\};$$

$$t^U_{2,2} + \max\{0; t^U_{1,2} - t^L_{2,1}\} \le t^L_{3,1};$$

$$t^U_{1,1} + t^U_{2,2} + \max\{t^U_{1,2}; t^U_{2,1}\} \le t^L_{1,1} + t^L_{2,1} + t^L_{3,1}.$$

Let processing set of jobs *J* start at time $\tau_0 = 0$, and at time $\tau_1 > \tau_0$ machine M_1 completed all the jobs before the next conflict pair of jobs. Without lose of generality, we assume that jobs (1, 2, …, j –1) are completed on machine M_1 in this order, jobs j and j +1 make next conflict pair of jobs, and job j +2 has to be processed optimally after this conflict pair of jobs. At time τ_1 we need to decide which job j or j +1 has to be processed next on machine M_1 in order to obtain an optimal schedule. It is clear that at time τ_1 the exact processing times of jobs (1, 2, …, j –1): $t_{1,1}, t_{2,1}, ..., t_{j-1,1}$, on machine M_1 are already known. We assume also that at time τ_1 the processing times of jobs {1, 2, …, j –1} on machine M_2 are also known. Thus, at time τ_1 the initial part $(t_{1,1}, t_{1,2}, ..., t_{j-1,1}, t_{j-1,2})$ of the vector $t \in T$ of the job processing times is already known.

Let c_1 denote the completion time of all jobs (1, 2, …, j –1) on machine M_1 and c_2 denote the completion time of all jobs (1, 2, …, j –1) on machine M_2.

Theorem 4 The o*rder $j \to j+1$ of conflict pair of jobs $j \in J$ and $(j+1) \in J$ is optimal for processing jobs of set J if at least one of the following seven conditions holds:*

$$c_1 + t^U_{j1} \le c_2; \qquad c_1 + t^U_{j1} + t^U_{j+1,1} \le c_2 + t^L_{j2}; \qquad (1)$$

$$c_1 + t^L_{j1} > c_2; \qquad t^L_{j+1,1} + t^L_{j+2,1} \ge t^U_{j2} + t^U_{j+1,2}; \qquad t^U_{j+1,1} \le t^L_{j2}; \qquad (2)$$

$$t^L_{j+1,1} > t^U_{j2}; \qquad c_1 + t^L_{j1} + t^L_{j+1,1} \ge c_2 + t^U_{j2}; \qquad t^L_{j+2,1} \ge t^U_{j+1,2}; \qquad (3)$$

$$c_1 + t^L_{j1} \le c_2; \qquad c_1 + t^U_{j1} > c_2; \qquad t^U_{j+1,1} \le t^L_{j2}; \qquad t^L_{j+1,1} + t^L_{j+2,1} \ge t^U_{j2} + t^U_{j+1,2}; \qquad (4)$$

$c_1 + t_{j1}^L + t_{j+1,1}^L \le c_2 + t_{j2}^U$; $\quad c_1 + t_{j1}^U + t_{j+1,1}^U > c_2 + t_{j2}^L$; $\quad t_{j+1,1}^L > t_{j2}^U$; $\quad t_{j+2,1}^L \ge t_{j+1,2}^U$; (5)

$c_1 + t_{j1}^L \le c_2$; $\quad c_1 + t_{j1}^L + t_{j+1,1}^L \le c_2 + t_{j2}^U$; $\quad c_1 + t_{j1}^U > c_2$; (6)

$t_{j+1,1}^U > t_{j2}^L$; $\quad t_{j+1,1}^L + t_{j+2,1}^L \ge t_{j2}^U + t_{j+1,2}^U$; $\quad t_{j+2,1}^L \ge t_{j+1,2}^U$;

$c_1 + t_{j1}^L > c_2$; $\quad t_{j+1,1}^U > t_{j2}^L$; $\quad t_{j+1,1}^L \le t_{j2}^U$; $\quad t_{j+1,1}^L + t_{j+2,1}^L \ge t_{j2}^U + t_{j+1,2}^U$. (7)

The experiments on PC have been realized for calculating percentage of the number of conflict pair of jobs resolved due to Theorem 3 and Theorem 4 (Table 3) and percentage of instances solvable exactly due to Theorems 3 and 4 (Table 4). It is clear that only serious of instances which has non-zero value in Table 2 were considered in Tables 3 and 4 (each serious with zero value in Table 2 is indicated by symbol – in Table 3 and in Table 4). For each series we generated and tested 1000 instances with no more than one conflict pair of jobs at any time (for each combination of *n* and *L* under consideration). To this end, we used conditions of Theorem 2.

We tested problems with $n \in$ {5, 10, 15, ..., 50} jobs, integer processing times with lower and upper bounds uniformly distributed in the range [10,1000], and $L \in$ {1, 2, 3, ..., 15}. We restricted the experiments by $n \le 50$ due to the results given in Table 2. For each job, the lower bound of the job processing times was randomly generated in the range [10,1000] and the upper bound of the job processing times was computed as follows: $t_{jm}^U = t_{jm}^L (1 + L\% / 100\%)$.

Table 3. Percentage of the number of conflict resolutions due to Theorem 3 and Theorem 4

n\L	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	73.53	71.21	72.89	70.73	69.91	69.19	69.03	70.32	68.88	67.25	68.28	65.96	66.16	65.17	63.47
10	92.32	91.8	92.76	91.46	91.96	93.33	90.83	91.08	91.38	90.64	90.67	90.54	90.71	90.07	91
15	96.97	97.41	96.85	96.13	97.12	96.89	96.31	97.12	96.8	97.04	97.31	96.14	96.62	95.96	96.07
20	98.3	97.78	98.17	98.13	97.52	97.86	98.17	98.19	98.18	98.08	97.84	97.65	98.17	97.67	97.56
25	98.55	99.04	99.07	98.75	98.73	98.57	98.5	98.59	99	98.73	98.63	98.55	98.68	98.52	98.19
30	99.2	99.18	99.03	99.11	99.01	99.03	99.13	99.11	98.82	-	-	-	-	-	-
35	98.76	99.22	99.34	99.28	99.34	99.48	99.2	-	-	-	-	-	-	-	-
40	99.36	99.45	99.57	99.63	99.47	-	-	-	-	-	-	-	-	-	-
45	99.42	99.72	99.57	99.57	99.42	-	-	-	-	-	-	-	-	-	-
50	99.3	99.51	99.49	-	-	-	-	-	-	-	-	-	-	-	-

Table 4. Percentage of the instances with all conflicts being resolved due to Theorem 3 and Theorem 4

n \ L	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	78.3	75.6	77.5	75.5	74.8	73.8	73.6	74.4	74	72.3	73	70.3	71.6	70	69
10	92.4	91.6	92.4	90.7	90.9	92.2	88.9	89	89.2	88	88.1	86.8	86.6	85.5	87.9
15	96.7	96.7	95.8	94.5	95.1	94.6	93.3	94.2	92.7	93.6	93.6	90.5	91.6	89.1	88.9
20	97.8	96.5	96.6	95.9	94.1	94	94.7	94.8	93.6	93	92.4	90.5	92.1	89.9	90.2
25	97.9	98.1	97.6	96.3	95.6	94.3	93.3	93.4	94.9	93.3	92.2	91.5	92	90.7	88.1
30	98.4	98	97.2	96.1	95.7	94.9	94.4	94.8	94	91.5	-	-	-	-	-
35	97.4	97.4	97.2	96.3	95.9	96.4	94	-	-	-	-	-	-	-	-
40	98.4	97.7	97.7	97.6	96	-	-	-	-	-	-	-	-	-	-
45	98.2	98.6	97.1	96.5	94.5	-	-	-	-	-	-	-	-	-	-
50	97.5	97	95.9	-	-	-	-	-	-	-	-	-	-	-	-

From Tables 3 and 4 it follows that the most randomly generated instances of the uncertain problem $F2 \mid t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \mid C_{\max}$ (each of which has no more than one conflict pair of jobs at any time) are solved exactly during realization of the process due to sufficient conditions given in Theorems 3 and 4 (in spite of the uncertain job processing times before scheduling).

Conclusion

It is clear that in spite of uncertainty of the job processing times it is necessary to choose only one schedule for practical realization of the process. Theorem 1 allows obtain a dominant permutation for problem $F2 \mid t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \mid C_{\max}$ before realization of the process. Indeed, such a dominant schedule (if any) is the best one for any feasible realization of the job processing times. If condition of Theorem 1 does not hold, then one can use Theorem 2, Theorem 3, or Theorem 4 for constructing optimal schedule during realization of the process under consideration.

Clearly, this approach is useful if the level of uncertainty is low enough (the best results were obtained for the non-zero cases presented in Table 2). If level of uncertainty exceeds a certain threshold, then others approaches to problem $F2 \mid t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \mid C_{\max}$ outperform the approach based on Theorems 1 – 4. If there is no possibility to construct one dominant schedule for problem $F2 \mid t_{jm}^{L} \le t_{jm} \le t_{jm}^{U} \mid C_{\max}$ (i.e., conditions of Theorem 1 do not hold), it may be fruitful to construct more general schedule form on the basis of a partial strong order of jobs *J* (Theorem 2). Then one can consider the realization stage of a schedule for the flow-shop when a part of the schedule is already realized. Theorems 3 and 4 show how to use additional information about realized operations in order to obtain better solution than that constructed before scheduling. In such a case, a realistic solution process can be seen as consisting of *static* and *dynamic* phases. At the static phase, a scheduler can construct a family of the dominant permutations. At the dynamic phase of the decision-making, a scheduler has to select an appropriate schedule from such a family of the dominant permutations to react in real-time to the actual processing times of the already completed jobs.

Thus, our approach falls into the category of predictive-reactive scheduling. The static phase (based on Theorems 1 and 2) may be considered as predictive scheduling and dynamic phase (based on Theorems 3 and 4) may be considered as reactive scheduling (see [Aytud *et al.*, 2005; Gupta, Stafford, 2006]).

It is interesting to find sufficient conditions for choosing the unique permutation that is optimal for any feasible processing times of the remaining operations. If a scheduler cannot make right decision at time $\tau > 0$, he (she) has to use one of the solution policies, which does not guarantee to find an optimal schedule for any realization of the remaining job processing times. The solution policy may be optimistic or pessimistic (see [Aytud *et al.*, 2005; Shafransky, 2005]), or a scheduler can minimize objective function in average. For an uncertain problem it may be necessary to look for an optimal scheduling policy that stochastically minimizes the makespan [Ku, Niu, 1986; Pinedo, 1995]. To this end, it is necessary to obtain the reliable probability distributions for the random processing times. In general case, the choice of the job may also be based on minimization of possible loss of the objective function value.

The research was partially supported by INTAS (project 03-51-5501) and ISTC (project B-986). The authors would like to thank Ms. Natalja G. Egorova for the help in computational experiments.

Bibliography

[Aytug *et al.*, 2005] H. Aytug, M.A. Lawley, K. McKay, S. Mohan, and R. Uzsoy. Executing production schedules in the face of uncertainties: A review and some future directions. European Journal of Operational Research. 161, 86-110, 2005.

[Davenport, Beck, 2000] A.J. Davenport, and J.C. Beck A survey for techniques for scheduling with uncertainty. Unpublished, available from http://www.eil.toronto.ca/profiles/chris/chris.papers.html, 2000.

[Elmaghraby, Thoney, 2000] S. Elmaghraby, and K.A. Thoney. Two-machine flowshop problem with arbitrary processing time distributions. IIE Transactions, 31, 467-477, 2000.

[Gupta, Stafford, 2006] J.N.D. Gupta, and E.F. Stafford Jr. Flowshop scheduling research after five decades. European Journal of Operational Research. 169, 699-711, 2006.

[Jackson, 1956] J.R. Jackson. An extension of Johnson's results on job lot scheduling. Naval Research Logistic Quarterly, 3, 201-203, 1956.

[Jansen, Mastrolilli, Solis-Oba, 2005] K. Jansen, M. Mastrolilli, R. Solis-Oba. Approximation schemes for job shop scheduling problems with controllable processing times. European Journal of Operational Research. 167, 297-319, 2005.

[Johnson, 1954] S.M. Johnson. Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1, 61—68, 1954.

[Ku, Niu, 1986] P.S. Ku, and S.C. Niu. On Johnson's two-machine flow-shop with random processing times. Operations Research. 34, 130-136, 1986.

[Leshchenko, Sotskov, 2005] N.M. Leshchenko, and Yu.N. Sotskov. Two-machine minimum-length shop-scheduling problems with uncertain processing times. In: Proceedings of XI International Conference “Knowledge-Dialogue-Solution”, June 20-24, Varna, Bulgaria, 375-381, 2005.

[Pinedo, 1995] M. Pinedo. Scheduling: Theory, Algorithms, and Systems. Prentice-Hall, Englewood Cliffs. 1995.

[Slowinski, Hapke, 1999] R. Slowinski, and M. Hapke. Scheduling Under Fuzziness. Physical-Verlag, Heidelberg, New York. 1999.

[Shafransky, 2005] Ya.M. Shafransky. Scheduling problems with uncertain parameters: research directions and some results. Informatika. 3, 5-15, 2005 (in Russian).

Authors' Information

Natalja M. Leshchenko – United Institute of Informatics Problems of National Academy of Sciences of Belarus, Surganova str., 6, 220012, Minsk, Belarus; e-mail: leshchenko@newman.bas-net.by

Yuri N. Sotskov – Prof., DSc. United Institute of Informatics Problems of National Academy of Sciences of Belarus, Surganova str., 6, 220012, Minsk, Belarus; e-mail: sotskov@newman.bas-net.by

ACCESS RIGHTS INHERITANCE IN INFORMATION SYSTEMS CONTROLLED BY METADATA

Mariya Chichagova, Ludmila Lyadova

Abstract: All information systems have to be protected. As the number of information objects and the number of users increase the task of information system’s protection becomes more difficult. One of the most difficult problems is access rights assignment. This paper describes the graph model of access rights inheritance. This model takes into account relations and dependences between different objects and between different users. The model can be implemented in the information systems controlled by the metadata, describing information objects and connections between them, such as the systems based on CASE-technology METAS.

Keywords: access control mechanisms, graph model, metadata, CASE-technology.

ACM Classification Keywords: D.2 Software engineering: D.2.0 General - Protection mechanisms.

Introduction

As information systems become larger and more complex, and as the number of their users increase, there are growing needs for methods that can simplify and even partly automate the process of access rights assignment.

The main problem of traditional access control mechanisms is that they don’t take into account the relations between information objects.