PLANNING IN TIME-MANAGEMENT AS SEQUENCING GIVEN JOBS WITH INTERVAL PROCESSING TIMES

Technical Report · April 2015

DOI: 10.13140/RG.2.1.4508.3043

Yuri N. Sotskov
United Institute Of Informatics Problems

Frank Werner
Otto-von-Guericke-Universität Magdeburg

PLANNING IN TIME-MANAGEMENT AS SEQUENCING GIVEN JOBS WITH INTERVAL PROCESSING TIMES

Yuri N. Sotskov

United Institute of Informatics Problems,
National Academy of Sciences of Belarus, Minsk, Belarus
e-mail: sotskov2005@hotmail.com

Natalia G. Egorova

United Institute of Informatics Problems,
National Academy of Sciences of Belarus, Minsk, Belarus
e-mail: natamog@ya.ru

Frank Werner

Faculty of Mathematics, Institute of Mathematical Optimization,
Otto-von-Guericke University, Magdeburg, Germany
e-mail: frank.werner@ovgu.de

April 29, 2015

Abstract: An automated system for planning and controlling working hours for a person (time-management) is considered. The main stage in planning of the working hours is the construction of an optimal schedule of the jobs given for an employee. We propose to use algorithms for an optimal processing of a set of jobs with uncertain (interval) processing times on a single machine. The optimality criterion under consideration is the minimization of the total weighted completion times of the given jobs. This criterion may be interpreted as a measure of the effectiveness of the employee to fulfill the given set of jobs. Due to the nature of the planning process of the jobs for a person, it is assumed that the time needed for processing any given job may be undetermined (unknown) until the completion time of the job. Therefore, in the construction of an optimal schedule in time-management, only an upper bound and a lower bound on the factual processing time are assumed to be known for each job to be processed by a person.

Keywords: time-management, scheduling, uncertainty, total weighted completion times

MSC: 90B35, 90B36

1 Introduction

In this paper, an automated system for planning and controlling working hours for a person (time-management) is considered. The main stage in planning of the working hours is the construction of an optimal schedule of the jobs given for an employee. We propose to use algorithms for an optimal processing of a set of jobs with uncertain (interval) processing times on a single machine [1]. The optimality criterion under consideration is the minimization of the sum of the weighted completion times $\sum w_i C_i$ of the given jobs. This criterion may be interpreted as a measure of the effectiveness of the employee to fulfill the given set of jobs [2].

Due to the nature of the planning process of the jobs for a person, it is assumed that the time needed for processing any given job may be undetermined (unknown) until the completion time of the job. Therefore, in the construction of an optimal schedule in time-management, only an upper bound and a lower bound on the factual processing time are assumed to be known for each job to be processed by a person.

The rest of the paper is organized as follows. In Section 2, it is shown that planning in time-management may be realized as sequencing the given jobs in a single-stage system with uncertain (interval) processing times of the jobs. A method based on the stability of an optimal schedule with respect to possible variations of the job processing times is described in Section 3. The stability box for an optimal permutation (schedule) for processing the given jobs is introduced and analyzed in Section 4. The optimality box for an optimal permutation for processing the jobs is discussed in Section 5. A computer application for time-management is briefly described in Section 6. Some concluding remarks are given in Section 7.

2 Construction of an optimal schedule in a single-stage system

The problem of scheduling jobs for a working day may be formulated as follows. For an employee, a set of jobs (tasks) is given, which is needed to be processed by the employee during his (her) working day. An important characteristic of scheduling in time-management is that the exact processing time of a human job may be unknown until the completion of the job. It is usual that only a lower bound and an upper bound on the processing time of the job may be known before scheduling for a man (woman). Indeed, it is usually possible to set a reasonable limit for the processing time of each human job. At the worst, one can set the lower bound to be a positive real number close to zero and the upper bound to be equal to the length of the planning horizon (e.g., to the length of the working day). Therefore, for time-management, we assume that at the scheduling stage, only a lower bound and an upper bound on the possible processing time for a job are reliably known.

Another important point is the scheduling goal for time-management. In this paper, we assume that the employee must perform a maximal number of the most important jobs from the given set during the working day. If the employee will not have finished or even not have started some jobs from the given set, these jobs must not be important compared to the jobs completed during the working day.

Traditionally in scheduling theory, a three-field notation $\alpha|\beta|\gamma$ is used to identify the scheduling problem, where α determines the processing system, β describes the properties of the processing system, and γ is the optimality criterion. Now, we can formulate the problem under consideration in terms of scheduling theory as follows.

Each job $J_i \in J$, $J=\{J_1, J_2, ..., J_n\}$, $n \geq 2$, has to be processed on a single machine. A weight $w_i > 0$ is assigned to any job $J_i \in J$, which characterizes the importance of an early completion of the job J_i. All the jobs are available for processing on the machine from the initial time $t = 0$ of the planning horizon (the working day in our case). The factual time p_i needed for processing the job $J_i \in J$ on the machine can be equal to any real number, which is bounded by a lower bound $p_i^L > 0$ and an upper bound $p_i^U \geq p_i^L$, these bounds being known prior to scheduling. The laws of the probability distributions of the random processing times of the jobs $J_i \in J$ are assumed to be unknown before scheduling. When implementing the constructed schedule, the processing time p_i of the job J_i may remain unknown until the completion time of the job J_i. It is also assumed that each job $J_i \in J$ should be processed on the machine within a time $p_i \in [p_i^L, p_i^U]$ without any interruption.

This uncertain problem of constructing an optimal schedule (an optimal permutation) for processing the job set J, in which the weighted sum $\sum_{i=1}^{n} w_i C_i$ of the completion times has the smallest value, is denoted as $1|p_i^L \leq p_i \leq p_i^U|\sum w_i C_i$ in scheduling theory. If the lower bound $p_i^L > 0$ and the upper bound p_i^U on the processing time of each job are equal, i.e., $p_i^L = p_i^U$, then the uncertain problem $1|p_i^L \leq p_i \leq p_i^U|\sum w_i C_i$ turns out to be the deterministic problem $1||\sum w_i C_i$, which is well-studied in the OR literature and may be solved in polynomial time. In the paper [3], it was proven that it requires $O(n \log n)$ elementary operations to construct a solution (an optimal schedule) for the deterministic problem $1||\sum w_i C_i$, if the following necessary and sufficient conditions for the optimality of a schedule are used for constructing an optimal permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n})$ of the jobs $J_{k_r} \in \{J_{k_1}, J_{k_2}, \ldots, J_{k_n}\} = \{J_1, J_2, ..., J_n\}$:

$$\frac{w_{k_1}}{p_{k_1}} \geq \frac{w_{k_2}}{p_{k_2}} \geq \ldots \geq \frac{w_{k_n}}{p_{k_n}} \tag{1}$$

In the general case of the problem $1|p_i^L \leq p_i \leq p_i^U|\sum w_i C_i$, the set of all possible vectors $p=\{p_1, p_2, ..., p_n\}$ of the job processing times will be denoted as follows:

$$T = \{p \mid p \in R_+^n,\ p_i^L \leq p_i \leq p_i^U,\ i \in \{1,2,\ldots,n\}\} = \times_{i=1}^{n}[p_i^L, p_i^U],$$

where R_+^n denotes the set of all non-negative real vectors with the dimension n. Hereinafter, $\times_{i=1}^{n}[p_i^L, p_i^U]$ denotes the Cartesian product of the set of the closed intervals of the possible job processing times. A fixed vector $p \in T$ of the processing times of the jobs $J_i \in J$ is called a scenario.

Next, we show that the problem $1 \mid p_i^L \le p_i \le p_i^U \mid \sum w_i C_i$ corresponds to the considered scheduling problem arising in time-management. As already noted, in time management, the employee must perform a maximal number of the most important jobs from the given set in order to increase his (her) income during the working day. So, in time-management a schedule $\pi_k = (J_{k_1} J_{k_2}, \ldots, J_{k_n})$ for the daily jobs will be optimal, if this schedule will provide a maximal income for the employee.

Since in scheduling theory the optimization criteria mainly consider the minimization of an objective function, we need to transform the above criterion $\sum w_i C_i$ to the one of maximizing the income of the employee. Let I denote the maximal income, which possibly can be received by the employee during his (her) working day. Then the minimization of the sum of the weighted completion times $\sum w_i C_i$ of the given jobs J is equivalent to the maximization of the income $I - \sum w_i C_i$. In the latter case, if the makespan $C_{\max} = \max\{C_i \mid J_i \in J\}$ for an optimal permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n})$ will be greater than the length C^d of the working day, $C_{\max} > C^d$, then the jobs from the corresponding final part $(J_{k_{r+1}}, J_{k_{r+2}}, ..., J_{k_n})$ of the schedule $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_r}, J_{k_{r+1}}, J_{k_{r+2}}, ..., J_{k_n})$ will be deleted from the daily schedule, if the following two inequalities hold:

$$C_{k_1} + C_{k_2} + ... + C_{k_r} \le C^d \text{ and } C_{k_1} + C_{k_2} + ... + C_{k_r} + C_{k_{r+1}} > C^d .$$

As it follows from (1), if the employee will not have finished or even not have started the jobs $\{J_{k_{r+1}}, J_{k_{r+2}}, ..., J_{k_n}\}$ from the given set J, these jobs will not be important compared to the jobs $\{J_{k_1}, J_{k_2}, \ldots, J_{k_r}\}$ completed during the working day.

As already noted, it was assumed that all the jobs J have to be processed without interruptions. Of course in real life, the staff can take breaks during their works, but this can only lead to an increase of the processing time for the planned job (the employee will need to tune in this job again after each interruption of this job). Moreover, such an interruption for any scenario cannot reduce the optimal (minimal) value of the objective function $\sum w_i C_i$ for the problem $1 \| \sum w_i C_i$ with this fixed scenario. We assume also that all the processing times p_i for all jobs $J_i \in J$ are uncertain, because there is always (at least) an error in computing the factual processing time p_i of the job $J_i \in J$, i.e., we assume that in time-management the strict inequality $p_i^L < p_i^U$ holds for each job $J_i \in J$.

Summarizing, we can argue that to find an optimal schedule (optimal permutation) for processing the jobs $J_i \in J$ in time-management, it is possible to use the results obtained in scheduling theory for the uncertain problem $1 \mid p_i^L \le p_i \le p_i^U \mid \sum w_i C_i$. In what follows, some results obtained for the problem $1 \mid p_i^L \le p_i \le p_i^U \mid \sum w_i C_i$ are presented. These results may be used in real time-management.

3 A method based on the stability of an optimal schedule

For the problem $1|p_i^L \le p_i \le p_i^U|\gamma$ with uncertain processing times of the jobs, there is usually no single permutation for processing all the jobs from the set J, which would remain optimal under all scenarios from the set T. Let $S=\{\pi_1,\pi_2,...,\pi_{n!}\}$ be the set of all permutations π_k determining the orders for processing the jobs from the set J: $\pi_k=(J_{k_1},J_{k_2},...,J_{k_n})$. The set of all permutations $S=\{\pi_1,\pi_2,...,\pi_{n!}\}$ has the cardinality $|S|=n!$. As a solution to the uncertain problem $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$, we will denote a minimal (with respect to an inclusion) set of the permutations respecting to the following definition given in the book [1].

Definition 1 *The set of permutations (semiactive schedules) $S(T)\subseteq S$ is called a minimal dominant set for the problem $\alpha|p_i^L \le p_i \le p_i^U|\gamma$, if the following two conditions hold:*

(a) for any fixed scenario $p\in T$, the set $S(T)$ contains at least one permutation of the jobs J, which is optimal for the deterministic problem $\alpha||\gamma$ with the scenario p,

(b) no proper subset of the set $S(T)$ possesses property (a).

The cardinality $|S(T)|$ of a minimal dominant set $S(T)$ can be viewed as a measure of the uncertainty of the problem $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$. In particular, in the case of the smallest value $|S(T)|=1$, the minimal dominant set for the problem $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ is a singleton $\{\pi_k\}=S(T)$, which is a solution (i.e., an optimal schedule) for the deterministic analogue $1||\sum w_iC_i$ of the original uncertain problem with any fixed scenario $p\in T$.

The minimal dominant set $S(T)$ can be constructed for an uncertain problem $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ based on the following dominance relation that can be efficiently (i.e. polynomially) determined on the set J of jobs [1].

Definition 2 *Job $J_u\in J$ dominates job $J_v\in J$ with respect to T, if there exists a minimal dominant set $S(T)$ for the problem $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ such that job J_u precedes job J_v in any permutation from the set $S(T)$.*

In the paper [4], the following statements for the uncertain problem $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ have been proven.

Theorem 1 *For the uncertain problem $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$, job J_u dominates job J_v with respect to T if and only if the following inequality holds:*

$$\frac{w_u}{p_u^U} \ge \frac{w_v}{p_v^L}.$$

Theorem 2 *For the existence of a minimal dominant singleton $S(T)=\{\pi_k\}=\{(J_{k_1},J_{k_2},\ldots,J_{k_n})\}$ for the problem $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$, it is necessary and sufficient that the following inequalities hold:*

$$\frac{w_{k_1}}{p^U_{k_1}} \ge \frac{w_{k_2}}{p^L_{k_2}}; \quad \frac{w_{k_2}}{p^U_{k_2}} \ge \frac{w_{k_3}}{p^L_{k_3}}; \quad \ldots \quad \frac{w_{k_{n-1}}}{p^U_{k_{n-1}}} \ge \frac{w_{k_n}}{p^L_{k_n}}.$$

T h e o r e m 3 *Let inequality $p_i^L < p_i^U$ hold for each job $J_i \in J$. Then for the existence of a minimal dominant set $S(T)$ for the problem $1|\,p_i^L \le p_i \le p_i^U\,|\sum w_iC_i$ with the maximal cardinality $|S(T)| = n!$, it is necessary and sufficient that the following inequality holds:*

$$\max\left\{\frac{w_i}{p_i^U} \,\middle|\, J_i \in J\right\} < \min\left\{\frac{w_i}{p_i^L} \,\middle|\, J_i \in J\right\}.$$

In applying the method based on the stability of an optimal schedule to solve the uncertain problem $1|\,p_i^L \le p_i \le p_i^U\,|\sum w_iC_i$, the following question remains still open: If the condition of Theorem 2 does not hold, i.e., a minimal dominant set $S(T)$ consists of more than one permutation, which permutation from the minimal dominant set $S(T)$ should be chosen for a practical implementation of the schedule? In the papers [6, 7], it was proposed to use a schedule based on the permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S(T)$ with the largest stability box for the uncertain problem $1|\,p_i^L \le p_i \le p_i^U\,|\sum w_iC_i$ for a practical implementation in time-management.

4 The stability box for an optimal permutation for processing the job set

By analogy with the definition given in [6], we define the stability box $SB(\pi_k, T)$ for an optimal permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$. To give the definition of the stability box, we need the following notations:

$$J^-[k_i] = \{J_{k_1}, J_{k_2}, \ldots, J_{k_{i-1}}\} \text{ and } J^+[k_i] = \{J_{k_{i+1}}, J_{k_{i+2}}, \ldots, J_{k_n}\}.$$

Let S_{k_i} denote the set of permutations $(\pi(J^-[k_i]), J_{k_i}, \pi(J^+[k_i])) \in S$ of the job set J, $\pi(J')$ being a permutation of the jobs $J' \subset J$. Let $1|\,p\,|\sum w_iC_i$ denote the deterministic problem $1||\sum w_iC_i$ with the scenario $p \in T$.

D e f i n i t i o n 3 *Let $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$ be a permutation of the jobs, which is optimal for at least one possible scenario $p \in T$. The maximal segment $[l_{k_i}, u_{k_i}] \subseteq [p^L_{k_i}, p^U_{k_i}]$ is called the maximal variation of the processing time J_{k_i} in the permutation π_k, if for any permutation $\pi_e \in S_{k_i}$ and any scenario $p = (p_1, p_2, \ldots, p_n) \in T$, for which the permutation π_e is optimal, the permutation π_k remains optimal for any scenario*

$$p' \in \times_{j=1}^{k_i-1}[p_j, p_j] \times [l_{k_i}, u_{k_i}] \times_{j=k_i+1}^{n} [p_j, p_j], \qquad (2)$$

provided that for any scenario $p'' = (p''_1, p''_2, \ldots, p''_n) \in T$, $p''_{k_i} \in [l_{k_i}, u_{k_i}]$, there exists an optimal permutation $\pi_v \in S_{k_i}$ for the instance $1|\,p''\,|\sum w_iC_i$. Let N_k denote the set of indexes i of all *jobs $J_i \in J$ with non-empty maximal variations of their processing times in the permutation. The Cartesian product*

$$SB(\pi_k, T) = \times_{k_i \in N_k} [l_{k_i}, u_{k_i}] \subseteq T$$

is called the stability box of the permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$. If there does not exist a scenario $p \in T$, such that the permutation π_k is optimal for the instance $1|p|\sum w_i C_i$, then it is assumed that $SB(\pi_k, T) = \varnothing$.

The dimension of the stability box $SB(\pi_k, T)$ is equal to the cardinality $|N_k|$ of the set N_k of the job indexes, the processing times of which have non-empty maximal variations. The cardinality $|N_k|$ of the set N_k is an important characteristic of the stability box $SB(\pi_k, T)$, since it determines the largest number of jobs, the duration p' of which can be varied in the scenario p inside the closed interval (segment) $[l_{k_i}, u_{k_i}]$ with guaranteeing the optimality of the permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$. The other processing times $\{p'_{k_j} \mid k_j \in N \setminus N_k\}$ from the scenario p', which are contained in a subset of the possible scenarios (2), should remain the same as in the scenario p' (i.e., $p'_{k_j} = p_{k_j}$) to ensure the optimality of the permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$ (see Definition 3).

The relative volume $VolSB(\pi_k, T)$ of the stability box $SB(\pi_k, T)$ of the permutation π_k for processing the jobs J is calculated as the product of the fractions

$$\frac{(u_{k_i} - l_{k_i})}{(p^U_{k_i} - p^L_{k_i})}$$

for all jobs J_{k_i} with the maximal variations $[l_{k_i}, u_{k_i}]$ such that inequality $l_{k_i} < u_{k_i}$ holds:

$$VolSB(\pi_k, T) = \prod_{l_{k_i} < u_{k_i}} \frac{u_{k_i} - l_{k_i}}{p^U_{k_i} - p^L_{k_i}}.$$

For finding the stability box $SB(\pi_k, T)$, it is sufficient to calculate the maximal variations of all durations p_{k_i} for all indexes $i \in \{1, 2, \ldots, n\}$. The maximal segment $[d^-_{k_i}, d^+_{k_i}]$ of possible changes of the weight-to-process ratio for job J_{k_i}, which corresponds to the maximal variation $[l_{k_i}, u_{k_i}]$ of the processing time p_i, will be called the maximal variation of the weight-to-process ratio for the job J_{k_i}. The lower bound $d^-_{k_i}$ on the maximal variation of the weight-to-process ratio is calculated as follows:

$$d^-_{k_i} = \max\left\{\frac{w_{k_i}}{p^U_{k_i}}, \max_{i < j \le n}\left\{\frac{w_{k_j}}{p^L_{k_j}}\right\}\right\}, i \in \{1, 2, \ldots, n\},$$

and the upper bound $d^+_{k_i}$ on the maximal variations of the weight-to-process ratio is calculated as follows:

$$d_{k_i}^{+} = \min\left\{\frac{w_{k_i}}{p_{k_i}^{L}},\ \min_{1\le j<i}\left\{\frac{w_{k_j}}{p_{k_j}^{U}}\right\}\right\}, i \in \{1,2,\dots,n\}.$$

The dimension $|N_k|$ of the stability box $SB(\pi_k, T)$ is equal to the number of jobs, for which inequality $d_{k_i}^{-} \le d_{k_i}^{+}$ holds. In the paper [6], several properties of the stability box have been proven and an $O(n\log n)$-algorithm for the construction of the stability box for a fixed permutation with n jobs has been developed using the proven properties of the stability box.

Let n_k denote the number of jobs, for which both the equality $d_{k_i}^{-} = d_{k_i}^{+}$ and the inequality $p_{k_i}^{L} < p_{k_i}^{U}$ hold. For the practical realization of scheduling the jobs in time-menagement, it is reasonable to use a permutation $\pi_t \in S$, which has

(a) the largest dimension $|N_t|$ of the stability box $SB(\pi_t, T)$,

(b) the minimal number n_k of the maximal variations with zero length (for the jobs having maximal variations with zero length, both the equality $d_{k_i}^{-} = d_{k_i}^{+}$ and the inequality $p_{k_i}^{L} < p_{k_i}^{U}$ must hold) among all permutations $\pi_k \in S$ having the largest dimension $|N_k| = |N_t|$ of their stability boxes $SB(\pi_k, T)$, and

(c) the largest relative volume $VolSB(\pi_t, T)$ of the stability box $SB(\pi_t, T)$ among all permutations $\pi_k \in S$ having the largest dimension $|N_k| = |N_t|$ of their stability boxes $SB(\pi_k, T)$ and the minimal number n_k of the maximal variations with zero length.

In the papers [7, 8], the claims about the properties of the stability box have been proven and an algorithm for constructing a permutation with the largest (in terms of the properties (a), (b), and (c)) stability box has been developed. The asymptotic complexity of the algorithm developed in [7] for constructing permutations with the largest stability box is $O(n\log n)$.

5 The optimality box for a permutation for processing the job set

The optimality box for a permutation π_k, which is the subset of the stability region and a superset of the stability box for the same permutation π_k, has been investigated in the paper [9].

Definition 4 *The maximal closed rectangular box*

$$OB(\pi_k, T) = \times_{k_i \in N_k} [\hat{l}_{k_i}, \hat{u}_{k_i}] \subseteq T$$

is called an optimality box for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ *with respect to* T*, if the permutation* π_k*, being optimal for the deterministic instance* $1|p|\sum w_i C_i$ *with a scenario* $p = (p_1, p_2, \dots, p_n) \in T$*, remains optimal for the deterministic instance* $1|p'|\sum w_i C_i$ *with any scenario*

$$p' \in [\hat{l}_{k_g}, \hat{u}_{k_g}] \times \{\times_{k_j \in N \setminus k_g} [p_{k_j}, p_{k_j}]\}.$$

If there does not exist a scenario $p \in T$ *such that the permutation* π_k *is optimal for the instance* $1 \mid p \mid \sum w_i C_i$ *, then it is assumed that* $OB(\pi_k, T) = \varnothing$.

For calculating the optimality box, one can calculate the stability box for the corresponding instance $1 \mid \hat{p}_i^L \le p_i \le \hat{p}_i^U \mid \sum w_i C_i$ with the reduced intervals of the job processing times $[\hat{p}_i^L, \hat{p}_i^U] \subseteq [p_i^L, p_i^U]$, which can be calculated using the following formulas (see Fig. 1):

$$\frac{w_i}{\hat{p}_i^L} = \min_{1 \le j \le i} \left\{ \frac{w_j}{p_j^L} \right\}, \quad \frac{w_i}{\hat{p}_i^U} = \max_{i \le j \le n} \left\{ \frac{w_j}{p_j^U} \right\},$$

for all indexes i, where $1 \le i \le n$.

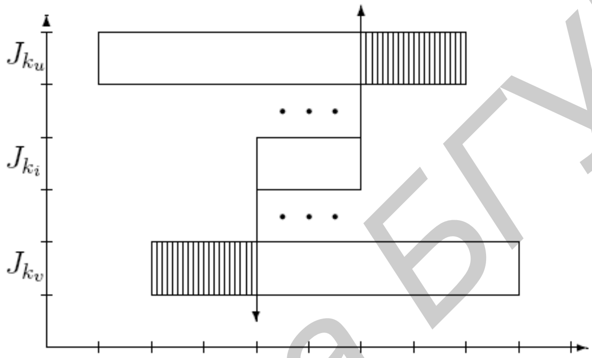

Figure 1. The reduced parts of the feasible closed intervals of the weight-to-process ratios are dashed.

The problem $1 \mid \hat{p}_i^L \le p_i \le \hat{p}_i^U \mid \sum w_i C_i$ with the reduced intervals of the job processing times $[\hat{p}_i^L, \hat{p}_i^U] \subseteq [p_i^L, p_i^U]$ is called the reduced problem. The following two theorems have been proven in the paper [9].

Theorem 4 *The optimality box for a permutation* π_k *of the jobs* J *for the problem* $1 \mid p_i^L \le p_i \le p_i^U \mid \sum w_i C_i$ *is equal to the optimality box for the permutation* π_k *of the jobs* J *for the corresponding reduced problem* $1 \mid \hat{p}_i^L \le p_i \le \hat{p}_i^U \mid \sum w_i C_i$.

Theorem 5 *For the reduced problem* $1 \mid \hat{p}_i^L \le p_i \le \hat{p}_i^U \mid \sum w_i C_i$, *the stability box* $SB(\pi_k, T)$ *for a permutation* π_k *of the jobs* J *is equal to the optimality box* $OB(\pi_k, T)$.

Using Theorem 4 and Theorem 5, an algorithm with the complexity $O(n)$ for constructing the optimality box $OB(\pi_k, T)$ has been developed in [9]. An experimental comparison of the dimensions and the relative volumes of the optimality box $OB(\pi_k, T)$ and the stability box $SB(\pi_k, T)$ of the optimal permutation of the jobs for randomly generated scenarios for the tested problems $1 \mid p_i^L \le p_i \le p_i^U \mid \sum w_i C_i$ with the order n, where $100 \le n \le 1000$, was conducted in [9]. Each series contained 100 instances with the same combination of the number n of jobs and the maximal possible error of the random processing times of the given jobs. As the computational experiments have shown, the optimality box had a considerably larger dimension than the stability box (by a factor of 28 on average), and the relative volume of the restricted optimality box was larger than the relative volume of the stability box (by a factor of 4.2 on average).

The selection of a job permutation with the largest optimality box can increase the probability of obtaining an optimal schedule despite of the fact, that the laws of the probability distributions of the random processing times are not known at the time of constructing a schedule for the uncertain problem $1|p_i^L \le p_i \le p_i^U|\sum w_i C_i$. The development of such an algorithm may be a subject of further research.

Summarizing the results of the above studies, we propose the following two-step algorithm for constructing a realizable permutation $\pi_k \in S(T)$ for an efficient processing of the job set J in the uncertain problem $1|p_i^L \le p_i \le p_i^U|\sum w_i C_i$.

1) Construct a minimal dominant set $S(T)$ for the uncertain problem $1|p_i^L \le p_i \le p_i^U|\sum w_i C_i$ under consideration.

2) If the constructed set $S(T)$ is a singleton, $|S(T)|=1$, then the single permutation from the set $S(T)$ is the desired one, otherwise, one must choose the job permutation $\pi_k \in S(T)$ with the largest stability box or the largest optimality box for the realization.

The development of an effective algorithm for searching a permutation $\pi_k \in S(T)$ with the largest optimality box and the development of algorithms for the correction of the minimal dominant set $S(T)$ constructed off-line, based on the additionally available information about the currently completed jobs, i.e., for an on-line modification of the set $S(T)$, are highly topical for time-management. Note that a similar investigation for the flowshop scheduling problem $F2|p_i^L \le p_i \le p_i^U|C_{\max}$ with two machines, interval processing times of the jobs and the criterion of minimizing the makespan $C_{\max}$ has been carried out in the paper [10].

6 Computer application for time-management

The method based on the stability of an optimal schedule and the corresponding algorithms will be used in the computer system for time-management, which was partly developed in [11]. The current version of this computer system uses new developments in the computer technology and new algorithms for scheduling jobs with uncertain (interval) processing times. The system will be realized as a distributed computer application, which can be used either by multiple users (in the hierarchy "supervisor (manager) – subordinate employers") or by a single independent user.

The computer application for time-management based on the model "supervisor (manager) – subordinate employers" is a distributed application of the type "client – server". This type of distributed applications is used because of the need to implement the interaction mechanism for several users on a single computer system, where each user should have access to the shared data to be able to modify the data and inform the other users (other members of the team). With a few modifications, this architecture "client – server" is used in the application both in the client part and in the server part of the application.

The application is developed based on the platform NET Framework 4.5 with the language of C# 5.0 and the following products and technologies: MS SQL Express 2012 (as the server database); Compact Edition (as the local database); Windows Communication Foundation (the technology of the service development in this case of the web services); IIS – Web server for deploying and running WCF services; WPF – a graphical subsystem for rendering user interfaces in Windows-based applications.

The basic logic of the server part will be presented by the algorithms for constructing schedules, monitoring schedules, as well as the functional user authentication, the user registration, and the user management. The client application for constructing a schedule for a person in the hierarchy "supervisor (manager) – subordinate employers" will provide different functional feasibilities depending on the user's role in the application, namely, the supervisor will have the complete set of the available functions, while the subordinate employer will have a limited set of the functions. The different application functionalities will be contained in separate modules that can be connected to the application when they will be needed for the current usage.

The common functionality for all users is as follows:

1) authentication and authorization;
2) user registration;
3) construction of a schedule for the user;
4) parameter settings;
5) notification service.

The common scenario for each user, when he (she) will start to use the application for the first time, is the user registration and the user authorization. These operations will be performed on the server via web services. After the user registration and authorization in the application, the functional configuration becomes available for him (her). Using this functional configuration of the application, the user can specify his (her) individual parameters for setting in the application. Both personal parameters (personal data) and general application parameters for time-management may be fixed to make the application more appropriate and effective for the user. The personal and general parameters, which are fixed by the user, will then be taken into account in the construction of the individual schedule.

A supervisor (manager) has the full functionality of the application. The functions of the application are as follows:

1) control of the users of the application;
2) input of the information about the jobs (this information will be used in the construction of the schedule and then in processing the jobs);
3) construction of the individual schedules;
4) the manual (or automatic) assignment of the jobs to the subordinate employers;
5) monitoring the implementation of the own schedule and the schedules of the subordinate employers.

The control of the users includes the view of their settings, time-management settings, the view of the jobs that are performed by the user, the password recovery, and the inclusion of new users into the system and removing the users from the application. In the future, generating and displaying the statistics associated with each user will be realized as well.

The schedule management module includes the function of constructing the schedules, the function of assigning jobs to the users, and the function of monitoring the realization of the schedule. This is the main module of the application. The process of constructing a schedule consists of the following steps:

1) receiving the input data from the database;
2) receiving the fixed settings, which have to be involved into the construction and realization of schedules for the manager and subordinate employers;
3) loading the libraries containing the implementation of the algorithms that perform the schedule construction.

These libraries can be developed and dynamically added to the system without the need of the development of special algorithms and their integration into the application. The algorithms for constructing the schedules can be changed at the stage of applying the personal settings. After constructing a schedule and assigning the jobs to the users to execute appropriate jobs, it is required to send a notification to the corresponding users.

At the current stage of the development of the application, the user's functional includes a minimal set of the operations required to perform their job schedules, namely: the view of the jobs, which are necessary to process including all information that is necessary for the user to process the job; the opportunity to arrange changes in the status of the job (job is waiting to be processed, job is processed, job is completed).

It should be noted that the use of this application will not be considered as a binding set of fixed instructions. Its main purpose is to give the user some tools to build a list of the assigned jobs in the order that would optimize the working hours. The mathematical apparatus, which is used for time-management, is designed to provide sufficient backgrounds for the correctness of the technology of time-management using the specifics of the jobs and the preferences of a particular user.

7 Conclusion

The problem of integrating algorithms of optimal scheduling and time-management principles within the framework of a unified system of time-management was considered. In scheduling theory, the planning jobs in time-management can be modeled as an optimization service system with a single machine and the criterion of minimizing the sum of the weighted completion times of the given jobs. Due to the nature of the planning process of the working time for a person, it was assumed that the processing time of each scheduled job can remain uncertain before the completion of the job. The known results, obtained for the solutions of the corresponding scheduling problems with uncertain processing times of the jobs were presented, and directions for further research in scheduling theory to be applied in time-management were discussed.

In the case when we have to consider multiple criteria, the set $S(T)$ can be reduced (see [5]). The list of possible optimization criteria for optimal schedules, which are appropriate to be considered in the construction of optimal schedules for time-management, were provided in [5]. However, for such a criterion, one needs to develop special algorithms for finding a minimal dominant set, which might be the subject of further research.

Acknowledgement

This work was partially supported by DAAD. The authors are indebted to Andrei A. Kasiankou for the qualified coding of the application for time-management.

References

1. Yu.N. Sotskov, N.Yu. Sotskova, T.-C. Lai, F. Werner, Scheduling under uncertainty: theory and algorithms, *RUE Publishing House «Belorusskaya nauka»*, 2010, 326 p.

2. N.G. Egorova, Yu.N. Sotskov, Minimizing the sum of the weighted completion times of the jobs with interval processing times, *Informatics*, No. 3, 2008, 5–16 (in Russian).

3. W.E. Smith, Various optimizers for single-stage production, *Naval Research Logistics Quarterly*, Vol. 3, No. 1, 1956, 59 – 66.

4. Yu.N. Sotskov, N.G. Egorova, T.-C. Lai, Minimizing total weighted flow time of a set of jobs with interval processing times, *Mathematical and Computer Modelling*, Vol. 50, No. 3 – 4, 2009, 556 – 573.

5. Yu.N. Sotskov, N.G. Egorova, N.M. Matsveichuk, E.A. Petrova, Models and program package to plan the working hours, *Informatics*, Vol. 4, 2007, 23 – 36 (in Russian).

6. Yu.N. Sotskov, T.-C. Lai, Minimizing total weighted flow time under uncertainty using dominance and a stability box, *Computers & Operations Research*, Vol. 39, 2012, 1271 – 1289.

7. N.G. Egorova, Yu.N. Sotskov, A.A. Kasiankou, A permutation with the largest stability box for processing jobs with interval processing times, *Informatics*, No. 4, 2012, 69 – 80 (in Russian).

8. Yu.N. Sotskov, N.G. Egorova, T.-C. Lai, F. Werner, The stability box in interval data for minimizing the sum of weighted completion times, *SIMULTECH 2011: Proc. Int. Conf. Simulat. Model. Method., Technological Application, Noordwijkerhout, Netherlands, 29 – 31 July 2011, SciTePress, Sci. Tecnol. Publicat., Portugal*, 2011, 14 – 23.

9. Yu.N. Sotskov, N.G. Egorova, Stability polyhedra of optimal permutation of jobs servicing, *Automation and Remote Control*, Vol. 75, No. 7, 2014, 1267 – 1282.

10. N.M. Matsveichuk, Yu.N. Sotskov, N.G. Egorova, T.-C. Lai, Schedule execution for two-machine flow-shop with interval processing times, *Mathematical and Computer Modelling*, Vol. 49, No. 5 – 6, 2009, 991 – 1011.

11. Yu.N. Sotskov, A.A. Kasiankou, Development of a computer application for optimal planning of working time, *Economics, modeling, forecasting*, Vol. 8, 2014, 138 – 148.